

Dimensional hyper-reduction of nonlinear finite element models via empirical cubature

J.A. Hernández^{a,b,*}, M.A. Caicedo^a, A. Ferrer^{a,b}

^a Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Technical University of Catalonia, Edificio C1, Campus Norte, Jordi Girona 1-3, Barcelona 08034, Spain

^b Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa, C/ Colom, 11, Terrassa 08222, Spain

Received 24 March 2016; received in revised form 18 July 2016; accepted 11 October 2016

Available online 20 October 2016

Abstract

We present a general framework for the dimensional reduction, in terms of number of degrees of freedom as well as number of integration points (“hyper-reduction”), of nonlinear parameterized finite element (FE) models. The reduction process is divided into two sequential stages. The first stage consists in a common Galerkin projection onto a reduced-order space, as well as in the condensation of boundary conditions and external forces. For the second stage (reduction in number of integration points), we present a novel cubature scheme that efficiently determines optimal points and associated positive weights so that the error in integrating reduced internal forces is minimized. The distinguishing features of the proposed method are: (1) The minimization problem is posed in terms of orthogonal basis vector (obtained via a *partitioned* Singular Value Decomposition) rather than in terms of snapshots of the integrand. (2) The volume of the domain is exactly integrated. (3) The selection algorithm need not solve in all iterations a nonnegative least-squares problem to force the positiveness of the weights. Furthermore, we show that the proposed method converges to the absolute minimum (zero integration error) when the number of selected points is equal to the number of internal force modes included in the objective function. We illustrate this model reduction methodology by two nonlinear, structural examples (quasi-static bending and resonant vibration of elastoplastic composite plates). In both examples, the number of integration points is reduced three order of magnitudes (with respect to FE analyses) without significantly sacrificing accuracy.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Keywords: Reduced-order model; Hyper-reduction; Optimized cubature; Finite elements; Singular Value Decomposition

1. Introduction

Generally speaking, model order reduction refers to any endeavor aimed at constructing a simpler model from a more complex one. The simpler model is usually referred to as the *reduced-order model* (ROM), while the more

* Corresponding author at: Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Technical University of Catalonia, Edificio C1, Campus Norte, Jordi Girona 1-3, Barcelona 08034, Spain.

E-mail address: jhortega@cimne.upc.edu (J.A. Hernández).

complex one is termed the full-order or high-fidelity model. This full-order model may be, for instance, – as is the case here – a finite element (FE) model.

The focus of the present paper is on the so-called *projection-based, reduced-order models*. The existence of such low-dimensional representations for a given parametrized finite element problem relies on the premise that the state variable can be accurately approximated by a linear combination of a few global basis vectors. The most common approach is to determine these basis vectors by applying some type of dimensionality reduction strategy (such as the Proper Orthogonal Decomposition, POD) over a so-called *training sample*. This sample is obtained by previously solving – in an *offline* stage – the full-order model for judiciously chosen values of the input parameters.

1.1. Approximation of nonlinear terms

In the general case of governing equations featuring terms that bear a nonaffine relationship with both the state variable and input parameters, the construction of an inexpensive low-dimensional model entails two sequential stages [1], namely: (1) projection onto the reduced basis, and (2) approximation of the nonlinear term. Once a basis matrix for the state variable is available, the projection stage is a standard operation consisting in introducing the approximation of the state variables in the governing equation, and then in posing the resulting equation in the space spanned by the basis vectors. This operation naturally leads to a significant reduction in the number of unknowns, and hence diminishes considerably the equation solving effort. However, in a general nonlinear case, the computational cost of evaluating the residual still depends on the size of the underlying finite element mesh—hence the need for a second reduction stage.

In contrast to the first reduction stage, which is more or less standard, the second stage of dimensionality reduction – Ryckelynck [2] coined the term *hyper-reduction* to refer to it – is far more challenging and still remains an issue of discussion in the model reduction community. In the following, we examine the various approaches encountered in the related literature to deal with this additional dimensionality reduction stage.

1.2. Classification of “hyper-reduction” methods

Let $\mathbf{F}^h \in \mathbb{R}^N$ denote¹ the full-order term bearing a general, nonaffine relationship with both the input variable and the state variable (in the context of this paper, $\mathbf{F}^h \in \mathbb{R}^N$ will be the vector of FE nodal internal forces). The state variable $\mathbf{z} \in \mathbb{R}^n$ is projected onto the reduced-order space \mathbb{R}^m by the matrix $\mathbf{\Phi} \in \mathbb{R}^{N \times n}$ ($\mathbf{F} = \mathbf{\Phi}^T \mathbf{F}^h$). Existing approaches for dealing with the approximation of \mathbf{F} can be broadly classified as *nodal vector approaches* and *integral approaches*.

1.2.1. Nodal vector approximation approaches (“gappy” data)

In this type of approaches, the approximation is carried out by replacing the finite element vector \mathbf{F}^h by a low-dimensional interpolant $\mathbf{F}^h \approx \mathbf{R}_F \mathbf{F}_z^h$, $\mathbf{R}_F \in \mathbb{R}^{N \times m}$ being the interpolation matrix, and \mathbf{F}_z^h the entries of \mathbf{F}^h corresponding to the degrees of freedom ($\mathbf{z} \subset \{1, 2, \dots, N\}$) at which the interpolation takes place. The interpolation matrix is obtained following the common procedure of computing a basis matrix for \mathbf{F}^h , and then determining a set of indices so that the error is minimized over a set of representative snapshots of \mathbf{F}^h . This set of interpolation indices can be determined *offline* using procedures such as the Empirical Interpolation Method (EIM) [3,4], the Best Points Interpolation Method (BPIM) [5], the Discrete BPIM [6] or the Missing Point Estimation Method [7]. The idea behind this vector approximation approach has its roots in the landmark work of Everson and Sirovich [8] for reconstruction of “gappy” data, and was historically the first proposal for dealing with nonlinear terms in model order reduction; it has been adopted by, among others, [1,9–14]. Alternatively, [2] proposes to bypass the construction of the low-dimensional interpolant and simply solve the balance equations at appropriately selected degrees of freedom (collocation).

¹ A word in notation is in order here. The superindex h is employed throughout the paper to denote finite element nodal quantities; bare symbols, on the other hand, are associated to reduced-order variables, that is, variables projected onto the reduced-order space. Likewise, FE and reduced-order dimensions are represented by upper-case and lower-case symbols, respectively. For instance, N and n denote the number of unknowns in the FE and reduced-order problems, respectively, whereas M and m represents the total number of integration points in the FE and reduced-order problem, respectively.

1.2.2. Integral approximation approaches

In a finite element context, \mathbf{F} can be regarded, not only as a projection of a large vector into a reduced-order space ($\mathbf{F} = \Phi^T \mathbf{F}^h$), but also as the result of integrating over the concerned domain $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) the corresponding reduced-order variable $\mathbf{f} = \Phi^T \mathbf{f}^h$ ($\mathbf{f}^h : \Omega \rightarrow \mathbb{R}^N$), i.e.:

$$\mathbf{F} = \Phi^T \int_{\Omega} \mathbf{f}^h d\Omega = \int_{\Omega} \mathbf{f} d\Omega. \quad (1)$$

Accordingly, the problem at hand can be also viewed as that of approximation of an *integral*, rather than approximation of a vector. In turn, this problem can be addressed by: (1) seeking a low-dimensional approximation of the integrand, or (2) approximating the integral itself as a weighted sum of the integrand evaluated at optimal sampling points.

1. *Interpolation of the integrand.* This type of approaches follows, in essence, the same procedure described for vector approximation approaches; the difference lies in that, rather than constructing an interpolant for the integral, in this case, it is the *integrand* that is subjected to approximation via interpolation, that is, if we make $\mathbf{f}(\mathbf{x}) \approx \sum_{g \in \mathbf{z}} \mathbf{R}_g(\mathbf{x}) \mathbf{f}(\mathbf{x}_g)$, where \mathbf{R}_g ($g \in \mathbf{z}$) stands for the interpolation functions, then we can write

$$\mathbf{F} = \int_{\Omega} \mathbf{f} d\Omega \approx \sum_{g \in \mathbf{z}} \left(\int_{\Omega} \mathbf{R}_g d\Omega \right) \mathbf{f}(\mathbf{x}_g) = \sum_{g \in \mathbf{z}} \mathbf{Q}_g \mathbf{f}(\mathbf{x}_g). \quad (2)$$

Hence, the integral can be approximated as a sum of the product of *matrix weights* \mathbf{Q}_g (which can be calculated offline) and the integrand evaluated at the interpolating points $\mathbf{x}_g \in \mathbf{z}$ —this set of points is to be chosen among the Gauss points of the underlying finite element mesh. This is the approach followed by [15–18].

2. *Cubature methods.* The last approaches to be discussed can be regarded as global *cubature methods*. To the best of the authors' knowledge, the first scheme of this type was proposed by An et al. (2009) [19] in the context of computer graphics applications, and was recently introduced in computational mechanics circles by Farhat and co-workers [20,21]. Following the classical recipe of Gaussian quadrature of polynomial functions, An et al. advocate to approximate the integral as a finite sum of *positive scalar weights* $\{\omega_g\}_{g=1}^m$ times the integrand evaluated at appropriately chosen sampling points:

$$\mathbf{F} \approx \sum_{g=1}^m \omega_g \mathbf{f}(\bar{\mathbf{x}}_g). \quad (3)$$

The strategy proposed by An et al. [19], and employed in successive refinements and extensions of the method [22–27], consists in determining, among the integration points of the FE mesh, a reduced set of points and associated *positive weights* so that the integration error is minimized over a set of representative samples of the integrand. The motivation behind constraining the weights to be *positive* scalars is that, in doing so, the contribution to the Jacobian matrix due to the nonlinear term inherits the spectral properties of its full-order counterpart. To put it simply, in a structural problem, if the FE stiffness matrix is symmetric and positive definite, so will be its reduced-order counterpart. It should be highlighted that this desirable attribute is not enjoyed by the other two approaches discussed above (at least in the context of standard Galerkin projection, see Ref. [28]). Indeed, interpolatory schemes ruin the symmetry and, depending on the location of the sampling points, may also destroy the positive definiteness of finite element stiffness matrices [17,14]. As a consequence, such schemes tend to be less robust than the finite element models they intend to approximate.

1.3. Goal of the paper and original contributions

All of the above described approaches are still in their infancy, and many theoretical and practical aspects still remain unaddressed. The goal of the present paper is to contribute to enhance and clarify some of these aspects in the particular case of *cubature-based ROMs*. Specifically, we propose a novel cubature approach – we call it for reasons that will become clear latter the *Empirical Cubature Method* – that addresses the following issues:

1. The cost associated to the greedy selection algorithm put forward by An et al. in their seminal paper [19] can prove unduly expensive in relatively large problems featuring high number of cubature points. To alleviate this (offline) computational cost, we propose to subject the integrand to a *dimensionality reduction process* (here, we use the Singular Value Decomposition, SVD). In doing so, the underlying minimization problem can be cast in terms of the first p dominant modes obtained from such a decomposition, rather than in terms of the set of P snapshots (in general, $p \ll P$, so the reduction in computational effort may prove substantial). Furthermore, we show that in problems in which the geometry is independent of the input parameters, the internal force modes can be alternatively obtained via an SVD of stress snapshots.
 - (a) The above reduction in offline computational cost is partially made at the expense of an additional SVD, which is, in its own right, a very memory demanding operation. To overcome possible memory bottlenecks, we have concocted a *partitioned SVD* that precludes the necessity of processing the whole snapshot matrix (and that is amenable to parallelization).
2. There are certain parametric ROMs in which the nonlinear term \mathbf{F} is by definition zero for any training configuration. This is the case, for instance, of the self-equilibrium problems appearing in multiscale hierarchical analysis, a research arena in which model reduction is particularly appealing, as evidences the intense research activity in the recent decade [17,29–36]. In this kind of problems, the minimization problem associated to the determination of optimal points and weights becomes ill-posed—it admits the trivial solution $\omega_g = 0$ ($g = 1, 2, \dots, m$) for any set of integration points. Based on an idea developed by the authors in Ref. [17] to deal with a similar problem in integrand-approximation approaches, we propose to decompose the snapshots into their component along the range of the integral operator, and its component along the nullspace. It is shown that this strategy, aside from eliminating the ill-posedness, leads to numerical cubature rules in which the volume, V , of the physical domain is *exactly integrated*, that is, $\sum_{g=1}^m \omega_g = V$.
3. Last but not least, we have devised a greedy selection algorithm able to find the absolute minimum (exact integration) of the optimization problem arising from the above mentioned decomposition. In particular, we have found that *exactly* integrating p integrand modes (plus the volume) requires $m = p + 1$ points. Furthermore, as opposed to existing approaches, the selection algorithm presented here need not solve, in all iterations, the expensive nonnegative least-squares problem to force the positiveness of the weights.

For the sake of concreteness, we have chosen as vehicle for exposing all the above ideas a classical nonlinear structural model in small strains, and consequently, the discussion that follows is couched in the terminology of structural finite element models. The state variable will be the vector of unknown nodal displacement and the nonlinear term

\mathbf{F}^h the vector of nodal internal forces. Nevertheless, the methodology is general and may be applied in many other contexts. For instance, in a nonlinear heat conduction problem, the state variable would be the vector of temperatures, and the nonlinear term \mathbf{F}^h the heat-flux vector.

To make the paper reasonably self-contained, we thoroughly describe, not only the proposed cubature scheme (Section 4), and the resulting hyper-reduced order model (Section 5), but also the parameterized finite element model (Section 2), as well as the reduced-order model arising from the first reduction stage (Section 3). Guidelines of how to deal with Dirichlet boundary conditions and external forces are also provided in Section 3. Likewise, recipes for computing displacement reduced basis specially suited for the numerical examples presented in Section 6 are explained in Appendix C—these recipes are also original contributions of this work. The previously mentioned partitioned SVD, on the other hand, is not in the main body of the paper but is relegated to Appendix B.

2. Parametrized finite element model

The parametrized nonlinear equation whose complexity we wish to reduce is the standard finite element, semi-discrete motion equation in its Lagrangian form:

$$\mathbf{M}^h \ddot{\mathbf{d}}^h(\boldsymbol{\mu}) + \mathbf{F}^h(\mathbf{d}^h, \mathbf{d}_0^h; \boldsymbol{\mu}) = \mathbf{F}_{ext}^h(\boldsymbol{\mu}) - \mathbf{M}_0^h \ddot{\mathbf{d}}_0^h(\boldsymbol{\mu}). \quad (4)$$

Here, $\mathbf{d}^h \in \mathbb{R}^N$ and $\mathbf{d}_0^h \in \mathbb{R}^{N_0}$ denote the vectors of unknown and prescribed nodal displacements, respectively. The dimensions of these two vectors (i.e., the number of unrestricted and restricted degrees of freedom N and N_0 , respectively) are assumed to remain constant along the analysis (constant Dirichlet boundaries). Superposed dots indicate material time derivative, i.e. $\ddot{\mathbf{d}}^h = \partial^2 \mathbf{d}^h / \partial t^2$. The set of input parameters is symbolically represented by $\boldsymbol{\mu}$,

Given $\mu \in \mathcal{D}$, $g^h(\mu) : [0, T] \rightarrow \mathbb{R}^{N_0}$, $u_0^h, v_0^h \in \mathbb{R}^N$, $F_{ext}^h(\mu) : [0, T] \rightarrow \mathbb{R}^N$, find $d^h : [0, T] \rightarrow \mathbb{R}^N$ such that

$$M^h \ddot{d}^h + F^h = F_{ext}^h - M_0^h \ddot{d}_0^h$$

where

$$F^h = \int_{\Omega} f^h d\Omega \approx \sum_{g=1}^M W_g f^h(x_g; \cdot)$$

subject to the Dirichlet boundary and initial conditions:

$$d_0^h = g^h, \quad d^h(0) = u_0^h, \quad \dot{d}^h(0) = v_0^h,$$

and the constitutive equations

$$\mathcal{H}(\sigma, d^h, d_0^h, \xi; \mu) \Big|_{x_g} = \mathbf{0}, \quad g = 1, 2, \dots, M$$

Box I. Statement of the finite element problem.

and the corresponding space by \mathcal{D} ; this set of inputs may encompass variations of the prescribed boundary conditions, body forces, material parameters, etc... (occasionally, for notational brevity, the variable time t will be also included in μ). $M^h \in \mathbb{R}^{N \times N}$ and $M_0^h \in \mathbb{R}^{N \times N_0}$ are the mass matrices that give the inertial forces caused by acceleration of unrestricted and restricted DOFs, respectively; they both are assumed to be independent of μ .

On the other hand, $F^h \in \mathbb{R}^N$ and $F_{ext}^h \in \mathbb{R}^N$ denote the vectors of nodal internal and external forces, respectively. For simplicity of exposition, F_{ext}^h is assumed to be independent of the state variable. Thus, the previously mentioned “offending”, nonlinear term requiring optimized cubature is the vector of nodal internal forces F^h . The FE integration rule employed to evaluate this vector (e.g., element-wise Gauss quadrature) will be characterized by the pairs $\{x_g, W_g\}_{g=1}^M$, $x_g \in \Omega$ being the position of the g th integration point, W_g the corresponding weight (this weight includes the Jacobian of the finite element containing the point), and M the total number of integration points; accordingly, we can compactly write

$$F^h = \int_{\Omega} f^h d\Omega \approx \sum_{g=1}^M W_g f^h(x_g) \quad (5)$$

where $f^h(x_g) : \mathcal{D} \rightarrow \mathbb{R}^N$ is the (sparse) internal force vector at the g th integration point. In a small strain setting, $f^h(x_g) = B^h(x_g) \sigma(x_g)$, $B^h(x_g) \in \mathbb{R}^{N \times s}$ being the classical (global) strain–displacement finite element matrix at point x_g , and $\sigma \in \mathbb{R}^s$ the stress vector ($s = 4, 6$, for plane stress/strain problems, and 3D problems, respectively). The nonlinearity between f^h and the state variable d^h may be of geometric nature (large strains) and/or material nature. Lastly, the constitutive relationship between the stress vector and both μ and the history of deformation at each integration point will be symbolically represented here by

$$\mathcal{H}(\sigma, d^h, d_0^h, \xi; \mu) \Big|_{x_g} = \mathbf{0}, \quad g = 1, 2, \dots, M \quad (6)$$

where ξ stands for the vector of internal variables. For future references, the statement of the finite element problem is summarized in Box I.

3. First reduction stage

3.1. Unknown nodal displacements

The whole idea of model reduction relies on the premise that, for any input parameter $\mu \in \mathcal{D}$, the displacement solution can be approximated by a set of n linearly independent basis vectors $\Phi_i \in \mathbb{R}^N$ ($i = 1, 2, \dots, n$), with $n \ll N$,

that is

$$\mathbf{d}^h(\boldsymbol{\mu}) \approx \boldsymbol{\Phi} \mathbf{d}(\boldsymbol{\mu}) \quad (7)$$

where $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1 \quad \boldsymbol{\Phi}_2 \quad \cdots \quad \boldsymbol{\Phi}_n]$ is the displacement basis matrix, and $\mathbf{d} \in \mathbb{R}^n$ the vector of (unknowns) reduced displacements—physically, the vector containing the amplitude of each displacement mode. There are various procedures for computing the basis matrix $\boldsymbol{\Phi}$. The most common one consists in, firstly, solving the full-order problem for representative values of the input parameters $\{\boldsymbol{\mu}^j\}_{j=1}^P$ ($\boldsymbol{\mu}^j \in \mathcal{D}$); then collecting the corresponding solutions in a snapshot matrix:

$$\mathbf{X}_d := [\mathbf{d}^h(\boldsymbol{\mu}^1) \quad \mathbf{d}^h(\boldsymbol{\mu}^2) \quad \cdots \quad \mathbf{d}^h(\boldsymbol{\mu}^P)]; \quad (8)$$

and, finally, extracting the dominant modes by processing this matrix using any type of dimensionality reduction. If the standard Singular Value Decomposition (SVD) is the strategy of choice, then $\mathbf{X}_d \approx \boldsymbol{\Phi} \boldsymbol{\Sigma}_\Phi \mathbf{V}_\Phi^T$, $\boldsymbol{\Sigma}_\Phi$ and $\mathbf{V}_\Phi \in \mathbb{R}^{P \times n}$ being the truncated matrices of singular values and right singular vectors, respectively. More sophisticated dimensionality reduction techniques can, of course, be used. For instance, in the example shown in Section 6.1, we use the elastic/inelastic SVD proposed by the authors in [17], that consists, roughly, in decomposing the ensemble of solutions in those obtained in the elastic range, and those obtained when inelastic processes come into play, and then perform separated SVDs for each ensemble. Likewise, in the example of Section 6.2, the basis matrix for displacements includes both SVD modes and the dominant natural vibration modes of the structure. Further details concerning these two alternative strategies are provided in Appendix C.

3.2. Input vectors

To construct a ROM whose complexity is completely independent of the size of the underlying FE mesh, it is necessary² to replace by low-rank representations, not only the solution vector \mathbf{d}^h , but also the *input* vectors of prescribed displacements (\mathbf{d}_0^h) and external forces³ (\mathbf{F}_{ext}^h). In fact, these two vectors usually admit *exact* approximations by linear combination of a few spatial basis vectors: if prescribed displacements are, say, uniform in space, just one spatial mode would suffice; if the spatial variation is linear, then two spatial modes would be needed, and so on. The coefficients in the linear combinations, on the other hand, can be obtained by *interpolation*. For instance, for the nodal

Register for free at <http://www.scipedia.com> to download the version without the watermark

$$\mathbf{d}_0^h(\boldsymbol{\mu}) = (\boldsymbol{\Xi} \boldsymbol{\Xi}_{\mathbf{b}}^{-1}) \mathbf{d}_0^h|_{\mathbf{b}}(\boldsymbol{\mu}) \quad (9)$$

where $\boldsymbol{\Xi} \in \mathbb{R}^{N_0 \times n_0}$ is the corresponding basis matrix, $\mathbf{b} \subseteq \{1, 2, \dots, N_0\}$ a set of n_0 admissible interpolation indices (admissible here means that the block matrix of $\boldsymbol{\Xi}$ corresponding to rows $\mathbf{b} = \{b_1, b_2, \dots, b_{n_0}\}$, denoted by $\boldsymbol{\Xi}_{\mathbf{b}}$, is invertible), and $\mathbf{d}_0^h|_{\mathbf{b}}$ the entries of \mathbf{d}_0^h corresponding to indices \mathbf{b} . By introducing the variables

$$\boldsymbol{\Phi}_0 := \boldsymbol{\Xi} \boldsymbol{\Xi}_{\mathbf{b}}^{-1} \quad \mathbf{d}_0 := \mathbf{d}_0^h|_{\mathbf{b}}. \quad (10)$$

Eq. (9) becomes expressible in a format similar to the approximation of the unknown displacement in Eq. (7), i.e.:

$$\mathbf{d}_0^h(\boldsymbol{\mu}) = \boldsymbol{\Phi}_0 \mathbf{d}_0(\boldsymbol{\mu}). \quad (11)$$

The basis matrix $\boldsymbol{\Xi}$ can be obtained by the same procedure used for the basis matrix of \mathbf{d}^h , that is, by collecting the vector of prescribed displacements for the training input parameters in a single matrix $\mathbf{X}_{d0} \in \mathbb{R}^{N_0 \times P}$, and then applying the SVD to get the matrix of left singular vectors. Once this matrix is at one's disposal, the set of interpolation indices \mathbf{b} can be easily determined by means of, for instance, the *Discrete Empirical Interpolation Method* (DEIM) [4].

² In many problems, computational savings achieved by compressing input vectors may be negligible in comparison with the savings obtained from approximating unknown displacement and internal forces, and therefore, this operation may be skipped.

³ For simplicity of exposition, we have assumed that the initial conditions \mathbf{u}_0^h and \mathbf{v}_0^h are independent of $\boldsymbol{\mu}$. Should such conditions be dependent on $\boldsymbol{\mu}$, the same procedure explained in the following is to be applied to \mathbf{u}_0^h and \mathbf{v}_0^h .

The interpolant of \mathbf{F}_{ext}^h can be constructed following the same approach, i.e., by making

$$\mathbf{F}_{ext}^h = (\boldsymbol{\Theta} \boldsymbol{\Theta}_c^{-1}) \mathbf{F}_{ext}^h \Big|_c \quad (12)$$

$\boldsymbol{\Theta} \in \mathbb{R}^{N \times n_f}$ and $\mathbf{c} \subset \{1, 2, \dots, N\}$ being the corresponding basis matrix and set of interpolation points, respectively.

3.3. Projection onto the reduced-order space

Introducing expressions (7) and (11) into the FE balance equation (4), and multiplying by $\boldsymbol{\Phi}^T$ (Galerkin projection), we get

$$(\boldsymbol{\Phi}^T \mathbf{M}^h \boldsymbol{\Phi}) \ddot{\mathbf{d}} + \boldsymbol{\Phi}^T \mathbf{F}^h = \boldsymbol{\Phi}^T \mathbf{F}_{ext}^h - (\boldsymbol{\Phi}^T \mathbf{M}_0^h \boldsymbol{\Phi}_0) \ddot{\mathbf{d}}_0. \quad (13)$$

Defining the *reduced mass matrices* $\mathbf{M} \in \mathbb{R}^{n \times n}$ and $\mathbf{M}_0 \in \mathbb{R}^{n \times n_0}$ by

$$\mathbf{M} := \boldsymbol{\Phi}^T \mathbf{M}^h \boldsymbol{\Phi} \quad \mathbf{M}_0 := \boldsymbol{\Phi}^T \mathbf{M}_0^h \boldsymbol{\Phi}_0 \quad (14)$$

and the *reduced vector of internal and external forces* ($\mathbf{F} \in \mathbb{R}^n$ and $\mathbf{F}_{ext} \in \mathbb{R}^n$, respectively) by

$$\mathbf{F} = \boldsymbol{\Phi}^T \mathbf{F}^h \quad \mathbf{F}_{ext} = \boldsymbol{\Phi}^T \mathbf{F}_{ext}^h \quad (15)$$

balance equation (13) can be rewritten as⁴

$$\mathbf{M} \ddot{\mathbf{d}} + \mathbf{F} = \mathbf{F}_{ext} - \mathbf{M}_0 \ddot{\mathbf{d}}_0. \quad (16)$$

Likewise, the initial and boundary conditions in the ROM become

$$\mathbf{d}_0 = \mathbf{g}_b^h(\boldsymbol{\mu}), \quad \mathbf{d}(0) = \mathbf{u}_0, \quad \dot{\mathbf{d}}(0) = \mathbf{v}_0 \quad (17)$$

where $\mathbf{u}_0 = \boldsymbol{\Phi}^T \mathbf{u}_0^h$ and $\mathbf{v}_0 = \boldsymbol{\Phi}^T \mathbf{v}_0^h$.

Lastly, substitution of Eq. (12) into the second equation in (15) leads to the low-dimensional representation of the reduced external force vector:

$$\mathbf{F}_{ext} = \mathbf{R}_{ext} \mathbf{F}_{ext}^h \Big|_c, \quad \text{where} \quad \mathbf{R}_{ext} = \boldsymbol{\Phi}^T (\boldsymbol{\Theta} \boldsymbol{\Theta}_c^{-1}). \quad (18)$$

3.4. Internal forces

The only term in Eq. (16) that still remains dependent on the complexity of the FE mesh (through the number of FE integration points M) is the reduced vector of internal forces $\mathbf{F} \in \mathbb{R}^n$; indeed, multiplying Eq. (5) by $\boldsymbol{\Phi}^T$ leads to

$$\mathbf{F} = \int_{\Omega} \boldsymbol{\Phi}^T \mathbf{f}^h \, d\Omega \approx \sum_{g=1}^M W_g \mathbf{f}(\mathbf{x}_g; \cdot), \quad (19)$$

where $\mathbf{f} := \boldsymbol{\Phi}^T \mathbf{f}^h$. To culminate the model-order reduction process, thus, we have to develop a more efficient integration rule for \mathbf{F} —one that acknowledges the fact that displacements, and therefore, strains, stresses and internal forces reside now in low-dimensional spaces. Approximation consistency considerations dictate that this last step is to be carried out in a second reduction stage, one in which the “reference” model is no longer the finite element model, but the ROM described heretofore. For the reader’s convenience and easy reference, the *offline* operations necessary to construct such a reduced-order problem, as well as the statement of the model itself, are summarized in [Boxes II](#) and [III](#), respectively.

⁴ It is at this point that the usefulness of our notational scheme becomes evident: to get the reduced-order balance equation (16), one just has to drop the superscript “ h ” in all the terms of the full-order equation (4)—bearing in mind the distinct definitions of the involved operators.

1. Solve the finite element problem (see [Box I](#)) for representative input parameters $\{\mu^i\}_{i=1}^P$. Store the resulting vectors of unrestricted nodal displacements (d^h), prescribed nodal displacements (d_0^h) and external nodal forces (F_{ext}^h) in the snapshot matrices X_d , X_{d0} , and $X_{f_{ext}}$, respectively.
2. Apply the SVD (or any other dimensionality technique alike) to matrices X_d , X_{d0} , and $X_{f_{ext}}$ to determine their respective basis matrices $\Phi \in \mathbb{R}^{N \times n}$, $\Xi \in \mathbb{R}^{N_0 \times n_0}$ and $\Theta \in \mathbb{R}^{N \times n_f}$.
3. Determine the interpolation indices $\mathbf{b} \subseteq \{1, 2, \dots, N_0\}$ and $\mathbf{c} \subseteq \{1, 2, \dots, N\}$ corresponding to basis matrices Ξ and Θ , respectively (using, for instance, the DEIM).
4. Compute the reduced-order matrices $\Phi_0 = \Xi \Xi_b^{-1}$, $M = \Phi^T M^h \Phi$, $M_0 = \Phi^T M_0^h \Phi_0$, $R_{ext} = \Phi^T \Theta \Theta_c^{-1}$, $u_0 = \Phi^T u_0^h$ and $v_0 = \Phi^T v_0^h$.

Box II. Offline operations (first reduction stage).

Given $\mu \in \mathcal{D}$, $g_b^h(\mu) : [0, T] \rightarrow \mathbb{R}^{n_0}$, $u_0, v_0 \in \mathbb{R}^n$, $F_{ext}^h|_c(\mu) : [0, T] \rightarrow \mathbb{R}^{n_f}$, find $d : [0, T] \rightarrow \mathbb{R}^n$ such that

$$M\ddot{d} + F = F_{ext} - M_0\ddot{d}_0$$

where

$$F = \int_{\Omega} f \, d\Omega \approx \sum_{g=1}^M W_g f(x_g; \cdot), \quad (f = \Phi^T f^h)$$

and $F_{ext} = R_{ext} F_{ext}^h|_c$, subject to the Dirichlet boundary and initial conditions:

$$d_0 = g_b^h, \quad \dot{d}(0) = u_0, \quad \ddot{d}(0) = v_0,$$

and to the constitutive equations

$$\mathcal{H}(\sigma, d, d_0, \xi; \mu)|_{x_g} = 0, \quad g = 1, 2, \dots, M$$

Box III. Statement of the reduced-order problem (without approximation of internal forces).

Register for free at <https://www.scipedia.com> to download the version without the watermark

4. Empirical cubature method (online stage)

In this section, we describe the formalisms concerning the problem of approximating the integral of the reduced vector of internal forces using the proposed *Empirical Cubature Method* (ECM). To originate our considerations from a general standpoint, we first deal with the case in which a *continuous* representation of the integrand is assumed to be available. Then, we move to the more practical scenario of only having at one's disposal a discrete representation of the integrand—that is, when the integrand is only known at the integration points of the FE mesh.

4.1. Standard “optimized” cubature scheme

As pointed out in Section 1.3, our proposal for efficiently integrating the vector of reduced internal forces draws on the *optimized cubature scheme* proposed by An et al. in Ref. [19]. We begin our discussion by briefly summarizing the main ingredients of this scheme. Suppose that the reduced-order problem described in [Box III](#) is solved for the set of input parameters $\{\mu^i\}_{i=1}^P$ (the same set employed in the first reduction stage). Let $f_I^j(x) = f_I(x, \mu^j)$ denote the I th component ($I = 1, 2, \dots, n$) of the integrand at point $x \in \Omega$ corresponding to the solution for input parameter μ^j ($j = 1, 2, \dots, P$). The idea put forward in Ref. [19] consists in approximating the integral of any $f_I^j : \Omega \rightarrow \mathbb{R}$ as the sum of positive, scalar weights multiplied by the function evaluated at appropriately chosen points, i.e.:

$$F_I^j = \int_{\Omega} f_I^j \, d\Omega \approx \sum_{g=1}^m \omega_g f_I^j(\bar{x}_g). \quad (20)$$

The positions of the integration points $\mathcal{Z} = \{\bar{\mathbf{x}}_g\}_{g=1}^m$ and their associated positive weights $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_m]^T$ ($\omega_g > 0$) are determined so that the integration error for all components and training samples is minimized:

$$(\boldsymbol{\omega}, \mathcal{Z}) = \arg \min_{\boldsymbol{\omega} \in \mathbb{R}_+^m, \mathcal{Z}_g \in \Omega} \sqrt{\sum_{I=1}^n \sum_{j=1}^P (e_I^j)^2} \quad (21)$$

where

$$e_I^j := \sum_{g=1}^m \omega_g f_I^j(\bar{\mathbf{x}}_g) - \int_{\Omega} f_I^j d\Omega. \quad (22)$$

Minimization problem (21) can be cast in matrix format as follows

$$(\boldsymbol{\omega}, \mathcal{Z}) = \arg \min_{\boldsymbol{\omega} \in \mathbb{R}_+^m, \mathcal{Z}_g \in \Omega} \|\mathbf{J}_{\mathcal{Z}} \boldsymbol{\omega} - \mathbf{b}\| \quad (23)$$

where $\|\cdot\|$ stands for the standard Euclidean norm, and

$$\mathbf{J}_{\mathcal{Z}} := \begin{bmatrix} f^1(\bar{\mathbf{x}}_1) & f^1(\bar{\mathbf{x}}_2) & \dots & f^1(\bar{\mathbf{x}}_m) \\ f^2(\bar{\mathbf{x}}_1) & f^2(\bar{\mathbf{x}}_2) & \dots & f^2(\bar{\mathbf{x}}_m) \\ \dots & \dots & \dots & \dots \\ f^P(\bar{\mathbf{x}}_1) & f^P(\bar{\mathbf{x}}_2) & \dots & f^P(\bar{\mathbf{x}}_m) \end{bmatrix}, \quad \mathbf{b} := \begin{bmatrix} \int_{\Omega} f^1 d\Omega \\ \int_{\Omega} f^2 d\Omega \\ \vdots \\ \int_{\Omega} f^P d\Omega \end{bmatrix}, \quad \mathbf{f}^j = \begin{bmatrix} f_1^j \\ f_2^j \\ \vdots \\ f_n^j \end{bmatrix}. \quad (24)$$

Notice that the g th column of $\mathbf{J}_{\mathcal{Z}} \in \mathbb{R}^{n \times m}$ contains the value of the integrand at the g th integration point for all training samples; the vector $\mathbf{b} \in \mathbb{R}^n$, on the other hand, is formed by arranging the *exact* integral of each function in a single column.

4.2. Dimensional reduction of the integrand

Register for free at <https://www.scipedia.com> to download the version without the watermark

Problem (23) represents a combinatorial, nonnegative minimization problem whose complexity depends, among other factors, on the total number of samples of the integrand (P). Especially in dynamic and/or history dependent problems, this number may prove inordinately high, for, in such cases, the time variable is considered as an input parameter, and, hence, one has to store the solution at all time steps, or a representative set of time steps. To alleviate the intrinsic computational burden of solving this optimization problem, we propose to *subject the integrand to a dimensionality reduction process* – in a manner similar to that explained for the displacement vectors in the first reduction stage –, and then pose the minimization problem (23) in terms of the (orthogonal) basis functions arising from this process, rather than in terms of the integrand itself.

Suppose that using some dimensionality reduction technique, we determine a set of $p \ll P$ basis functions for f_I^j so that

$$f_I^j(\mathbf{x}) \approx \sum_{i=1}^p \Lambda_i(\mathbf{x}) c_{iI}^j, \quad (j = 1, 2, \dots, P; \quad I = 1, 2, \dots, n) \quad (25)$$

where $\Lambda_i : \Omega \rightarrow \mathbb{R}$ stands for the i th basis function, and $c_{iI}^j \in \mathbb{R}$ is the corresponding coefficient in the approximation. Introducing the above into the expression for the integration error (22), we get

$$e_I^j = \left(\sum_{g=1}^m \omega_g \Lambda_i(\bar{\mathbf{x}}_g) - \int_{\Omega} \Lambda_i d\Omega \right) c_{iI}^j. \quad (26)$$

Since the coefficients are independent of the position, it follows that minimizing the integration error e_I^j over all training samples and components is equivalent to *minimizing the error incurred in approximating the integral of the*

basis functions Λ_i ($i = 1, 2, \dots, p$). This consideration leads to a minimization problem similar to problem (23), but in which the matrices $\mathbf{J}_{\mathcal{Z}}$ and \mathbf{b} are defined in terms of the orthogonal basis functions Λ_i ($i = 1, 2, \dots, p$) – rather than in terms of the samples of the integrand f_I^j –, i.e.:

$$\mathbf{J}_{\mathcal{Z}} := \begin{bmatrix} \Lambda_1(\bar{\mathbf{x}}_1) & \Lambda_1(\bar{\mathbf{x}}_2) & \cdots & \Lambda_1(\bar{\mathbf{x}}_m) \\ \Lambda_2(\bar{\mathbf{x}}_1) & \Lambda_2(\bar{\mathbf{x}}_2) & \cdots & \Lambda_2(\bar{\mathbf{x}}_m) \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_p(\bar{\mathbf{x}}_1) & \Lambda_p(\bar{\mathbf{x}}_2) & \cdots & \Lambda_p(\bar{\mathbf{x}}_m) \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \int_{\Omega} \Lambda_1 d\Omega \\ \int_{\Omega} \Lambda_2 d\Omega \\ \vdots \\ \int_{\Omega} \Lambda_p d\Omega \end{bmatrix}. \quad (27)$$

Thus, in subjecting the integrand to a dimensionality reduction process, the number of rows of both $\mathbf{J}_{\mathcal{Z}}$ and \mathbf{b} decreases from Pn to p (where $p \ll P$), and, consequently, so does the computational cost associated to solving the minimization problem.

4.3. Basis functions for the integrand: continuous case

4.3.1. Shortcoming of standard approaches

We now face the problem of developing a general strategy for determining the basis functions $\Lambda_i : \Omega \rightarrow \mathbb{R}$ for the integrand. In principle, one may use the standard strategy of directly applying the POD or techniques alike to the ensemble of snapshot of the integrand function $\{f_I^j(\mathbf{x})\}_{j=1}^P$. However, as pointed out in the introductory section, this strategy renders the minimization problem (23) ill-posed in cases in which the integral of such samples is zero for all values of the input parameters,⁵ i.e., when

$$\int_{\Omega} f_I^j d\Omega = 0 \quad (28)$$

for all $j = 1, 2, \dots, P$ and $I = 1, 2, \dots, n$. Indeed, since the basis functions are linear combination of snapshots, their integrals will be also zero, and consequently, $\mathbf{b} = \mathbf{0}$. Under such circumstances, minimization problem (23) becomes

$$(\boldsymbol{\omega}, \bar{\mathcal{Z}}) = \arg \min_{\boldsymbol{\omega} \geq 0, \bar{\mathcal{Z}}_g \in \Omega} \|\mathbf{J}_{\bar{\mathcal{Z}}} \boldsymbol{\omega}\|, \quad (29)$$

which possesses the trivial solution $\boldsymbol{\omega} = \mathbf{0}$ (for any set of integration points). Of course, this ill-posedness may be simply alleviated by replacing the non-negative constraint by each ω_g being greater than a certain (small) tolerance. However, the quality of this solution becomes sensible to the value of the chosen tolerance and, furthermore, it leads to integration schemes requiring more integration points than strictly necessary.

4.3.2. Expanded basis approach (EBA)

The approach adopted here to eliminate the above-mentioned ill-posedness is to use an “expanded basis” (in a sense that will become clear in the ensuing discussion) for the nonlinear term. This idea was introduced by the authors in Ref. [17] for dealing with problems of similar nature appearing in the context of schemes based on interpolation of the integrand. The gist of this general idea goes as follows. Suppose that the equations arising from the Galerkin projection (first reduction stage) are compactly written as

$$\mathbf{A}(s) = \mathbf{D}, \quad (30)$$

where s is the nonlinear term we wish to approximate, \mathbf{A} is a linear operator independent of both the input parameter and the state variable, and $\mathbf{D} \in \mathbb{R}^n$ the sum of the remaining terms of the governing equation. Let us assume for the

⁵ This situation is encountered in quasi-static problems with no external forces (only prescribed displacements).

sake of argument that $\mathbf{s} \in \mathbb{R}^M$, and that, therefore, \mathbf{A} admits a matrix representation, denoted also by \mathbf{A} . Then suppose that \mathbf{s} in Eq. (30) is replaced by the approximation $\mathbf{s} \approx \sum_{i=1}^p \boldsymbol{\Psi}_i c_i = \boldsymbol{\Psi} \mathbf{c}$:

$$\mathbf{A} \left(\sum_{i=1}^p \boldsymbol{\Psi}_i c_i \right) = \mathbf{D} \Rightarrow (\mathbf{A} \boldsymbol{\Psi}) \mathbf{c} = \mathbf{D} \quad (31)$$

$\{\boldsymbol{\Psi}_i\}_{i=1}^p$ being a set of basis vectors (and $\boldsymbol{\Psi} \in \mathbb{R}^{M \times p}$ the corresponding basis matrix). Expression (31) represents a system of n scalar equations that are *linear* in the coefficients $c_i = c_i(\boldsymbol{\mu}, \mathbf{d})$, although nonlinear, in general, in the state variable $\mathbf{d} \in \mathbb{R}^n$. It may be inferred from this fact, thus, that a *necessary* condition for Eq. (31) to have a unique solution in the state variable \mathbf{d} is that these n equations must be all linearly independent.⁶ This implies that the rank of the product of \mathbf{A} times the basis matrix $\boldsymbol{\Psi}$ must be equal to the number of unknowns n , i.e. $\text{rank}(\mathbf{A} \boldsymbol{\Psi}) = n$.

This necessary condition poses certain restrictions on the number and form of employed basis functions $\{\boldsymbol{\Psi}_i\}_{i=1}^p$. One of such restrictions is obvious: the number of modes for \mathbf{s} must be equal or greater than the number of unknowns ($p \geq n$), on the grounds that⁷

$$\text{rank}(\mathbf{A} \boldsymbol{\Psi}) \leq \min(\text{rank}(\mathbf{A}), \text{rank}(\boldsymbol{\Psi})) = \min(n, p). \quad (32)$$

Another (less obvious) condition, articulated and formulated by the authors in Ref. [17], is that the *span of the basis vectors must cover entirely the range of the operator \mathbf{A}* (denoted by $\mathcal{R}(\mathbf{A})$), or, to put it another way, the intersection of the range of the operator and the span of the chosen basis functions must be the range of the operator itself:

$$\mathcal{R}(\mathbf{A}) \cap \text{span}\{\boldsymbol{\Psi}_i\}_{i=1}^p = \mathcal{R}(\mathbf{A}). \quad (33)$$

The proof that this condition guarantees that $\text{rank}(\mathbf{A} \boldsymbol{\Psi}) = n$ follows from the (columnwise) orthogonal decomposition of $\boldsymbol{\Psi}$ into its components along the kernel (denoted by $\mathcal{N}(\mathbf{A})$) and range of \mathbf{A} :

$$\boldsymbol{\Psi} = \boldsymbol{\Psi}^R + \boldsymbol{\Psi}^N, \text{ with } \boldsymbol{\Psi}_i^R \in \mathcal{R}(\mathbf{A}), \boldsymbol{\Psi}_i^N \in \mathcal{N}(\mathbf{A}). \quad (34)$$

Indeed, since $\mathbf{A} \boldsymbol{\Psi}^N = \mathbf{0}$, we have that $\text{rank}(\mathbf{A} \boldsymbol{\Psi}) = \text{rank}(\mathbf{A} \boldsymbol{\Psi}^R)$. Furthermore, if Eq. (33) holds, then $\text{rank}(\boldsymbol{\Psi}^R) = n$. By elementary properties of the rank of product of matrices (see for instance Ref. [37]), it follows finally that $\text{rank}(\mathbf{A} \boldsymbol{\Psi}^R) = \text{rank}(\mathbf{A}) = n$.

If the basis functions are simply taken as a linear combination of snapshots of the concerned term \mathbf{s} evaluated at the solution, satisfaction of (33) becomes contingent on the particular form of the right-hand side \mathbf{D} . The worst conceivable scenario for its fulfillment is when $\mathbf{D} = \mathbf{0}$ for any input parameter $\boldsymbol{\mu} \in \mathcal{D}$ —which, incidentally, is the case at hand. Indeed, in such circumstances, any $\mathbf{f}(\boldsymbol{\mu}^j)$ lies in the kernel of \mathbf{A} , and therefore, the basis functions themselves will reside in such a subspace; since $\mathcal{N}(\mathbf{A})$ and $\mathcal{R}(\mathbf{A})$ are orthogonal subspaces, it follows that $\mathcal{R}(\mathbf{A}) \cap \text{span}\{\boldsymbol{\Psi}_i\}_{i=1}^p = \emptyset$ —hence the ill-posedness.

The proposal advocated by the authors in [17] to guarantee that condition (33) is invariably observed – regardless of the particular form of the right-hand side \mathbf{D} – is to construct the set of basis matrix by *expanding* the reduced basis set for the nonlinear term \mathbf{s} with a basis for $\mathcal{R}(\mathbf{A})$:

$$\underbrace{\{\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \dots, \boldsymbol{\Psi}_p\}}_{\text{Basis for } \mathbf{s}}, \underbrace{\{\boldsymbol{\Upsilon}_1, \boldsymbol{\Upsilon}_2, \dots, \boldsymbol{\Upsilon}_n\}}_{\text{Basis for } \mathcal{R}(\mathbf{A})}. \quad (35)$$

However, in the problem addressed in Ref. [17], the right-hand side term is invariably zero (self-equilibrium problem), and under such conditions, the set of vectors in Eq. (35) is linearly independent (since $\boldsymbol{\Psi}_i \cdot \boldsymbol{\Upsilon}_j = \mathbf{0}$). To extend the method for cases in which $\mathbf{D} \neq \mathbf{0}$, we propose here to exploit the orthogonal decomposition induced by the operator and construct the basis set as

$$\underbrace{\{\hat{\boldsymbol{\Psi}}_1, \hat{\boldsymbol{\Psi}}_2, \dots, \hat{\boldsymbol{\Psi}}_p\}}_{\text{Basis for } \hat{\mathbf{s}}}, \underbrace{\{\boldsymbol{\Upsilon}_1, \boldsymbol{\Upsilon}_2, \dots, \boldsymbol{\Upsilon}_n\}}_{\text{Basis for } \mathcal{R}(\mathbf{A})} \quad (36)$$

⁶ As argued in Ref. [17], in interpolatory approaches (when the coefficients c_i are determined by interpolation), failure to meet this necessary condition is conducive to rank deficient Jacobian matrices.

⁷ It is assumed that Eq. (30) is well-posed, and therefore, \mathbf{A} has full rank, $\text{rank}(\mathbf{A}) = n$.

where $\{\hat{\Psi}\}_{i=1}^p$ is a reduced basis for the orthogonal projection of s onto the kernel of A (denoted by \hat{s}). In doing so, $\hat{\Psi}_j \in \mathcal{N}(A)$, and hence $\langle \hat{\Psi}_j, \Upsilon_i \rangle = 0$ (here, $\langle \cdot, \cdot \rangle$ symbolizes the scalar product employed in the orthogonal decomposition of the pertinent space in $\mathcal{R}(A)$ and $\mathcal{N}(A)$).

4.3.3. Application of the EBA to the cubature problem

Let us now discuss the application of the method described in the foregoing to the problem at hand, which is to approximate $f_I : \mathcal{D} \times \Omega \rightarrow \mathbb{R}$ in equation

$$\int_{\Omega} f_I d\Omega = F_I, \quad I = 1, 2, \dots, n. \quad (37)$$

Comparison of the preceding equation with Eq. (30) indicates that, in this case, it is the *integral* itself that plays the role of the linear operator A . Thus, for purposes of constructing the reduced basis of f_I , we shall regard the integral as an operator \mathcal{I} that maps elements of the space of integrable functions on the domain $\Omega \subset \mathbb{R}^d$ (denoted by \mathcal{V}) to \mathbb{R} . Next we introduce the orthogonal decomposition, in the L_2 -norm, of \mathcal{V} into the range and kernel of \mathcal{I} , i.e.: $\mathcal{V} = \mathcal{N}(\mathcal{I}) \oplus \mathcal{R}(\mathcal{I})$. The kernel of the integral operator is formed by all integrable functions whose integral is zero:

$$\mathcal{N}(\mathcal{I}) = \left\{ g \neq 0 \in \mathcal{V} \mid \int_{\Omega} g d\Omega = 0 \right\}. \quad (38)$$

On the other hand, the range of \mathcal{I} is formed by those functions of \mathcal{V} that are orthogonal to $\mathcal{N}(\mathcal{I})$, that is

$$\mathcal{R}(\mathcal{I}) = \left\{ h \neq 0 \in \mathcal{V} \mid \forall g \in \mathcal{N}(\mathcal{I}), \int_{\Omega} g h d\Omega = 0 \right\}. \quad (39)$$

It readily follows from the preceding definition that the *range of the integral operator is the space of constant functions in Ω* ; thus, the dimension of $\mathcal{R}(\mathcal{I})$ is equal to 1, and a basis set for this space is formed by any constant function over Ω . Likewise, it is immediate to see that the projection of any sample of the integrand f_I^j ($I = 1, 2, \dots, n$, $j = 1, 2, \dots, P$) onto $\mathcal{N}(\mathcal{I})$ can be calculated by subtracting to f_I^j its *average* value over Ω , i.e.:

$$\hat{f}_I^j = f_I^j - \frac{1}{V} \int_{\Omega} f_I^j d\Omega, \quad I = 1, 2, \dots, n; \quad j = 1, 2, \dots, P \quad (40)$$

where $V = \int_{\Omega} d\Omega$ is the volume of the domain. Therefore, the desired set of basis functions for the integrand is to be constructed as the union of a basis set for the zero-average snapshots (40) (obtained via the POD, for instance) and a constant function (for instance, $g(\mathbf{x}) = 1, \forall \mathbf{x} \in \Omega$)

$$\underbrace{\{A_1, A_2, \dots, A_p\}}_{\text{Basis for } \hat{f}_I^j}, \quad \underbrace{\{1\}}_{\text{Basis for } \mathcal{R}(\mathcal{I})}. \quad (41)$$

Let us now discuss the implications of using the above basis set in the formulation of the minimization problem (23). Introducing (41) in the expressions (27) of $J_{\mathcal{Z}}$ and \mathbf{b} , we arrive at

$$J_{\mathcal{Z}} = \begin{bmatrix} \hat{J}_{\mathcal{Z}} \\ \mathbf{1}^T \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \hat{\mathbf{b}} \\ V \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ V \end{bmatrix} \quad (42)$$

where $\hat{J}_{\mathcal{Z}} \in \mathbb{R}^{p \times m}$ and $\hat{\mathbf{b}} \in \mathbb{R}^p$ are defined by

$$\hat{J}_{\mathcal{Z}} := \begin{bmatrix} A_1(\bar{\mathbf{x}}_1) & A_1(\bar{\mathbf{x}}_2) & \cdots & A_1(\bar{\mathbf{x}}_m) \\ A_2(\bar{\mathbf{x}}_1) & A_2(\bar{\mathbf{x}}_2) & \cdots & A_2(\bar{\mathbf{x}}_m) \\ \vdots & \vdots & \ddots & \vdots \\ A_p(\bar{\mathbf{x}}_1) & A_p(\bar{\mathbf{x}}_2) & \cdots & A_p(\bar{\mathbf{x}}_m) \end{bmatrix} \quad \hat{\mathbf{b}} = \begin{bmatrix} \int_{\Omega} A_1 d\Omega \\ \int_{\Omega} A_2 d\Omega \\ \vdots \\ \int_{\Omega} A_p d\Omega \end{bmatrix} = \mathbf{0} \quad (43)$$

whereas $\mathbf{1}^T = [1, 1, \dots, 1]$ (an all-ones row vector of dimension m). Substitution of the new definitions of $\mathbf{J}\mathbf{z}$ and \mathbf{b} in the expression of the objective function in problem (23) leads to

$$\|\mathbf{J}\mathbf{z}\boldsymbol{\omega} - \mathbf{b}\|^2 = \|\hat{\mathbf{J}}\mathbf{z}\boldsymbol{\omega}\|^2 + \|\mathbf{1}^T\boldsymbol{\omega} - V\|^2 = \|\hat{\mathbf{J}}\mathbf{z}\boldsymbol{\omega}\|^2 + \left(\sum_{g=1}^m \omega_g - V\right)^2. \quad (44)$$

Notice that the second term in the above objective function vanishes when the sum of the weights equal the volume of the domain. Therefore, aside from eliminating the ill-posedness of the standard approach, the proposed strategy leads to a minimization problem in which *the requirement that the sum of weights has to be as close as possible to the total volume of the domain is explicitly imposed*. It should be mentioned that neither the original cubature scheme of An. et al. [19], nor later related proposals (e.g. [20]), exhibits this remarkable feature.

Remark 4.1. It is interesting to note that the format exhibited by $\mathbf{J}\mathbf{z}$ and \mathbf{b} in the proposed approach (Eq. (42)) bears a close resemblance to their counterparts in classical Gaussian quadrature (see, for instance, page 143 in Ref. [38]). The reason of such a similarity is that they share a common attribute: the use of orthogonal basis functions. However, Gaussian formulae were originally conceived for one-dimensional domains (quadrature) and polynomial functions, while our approach can be applied to any multidimensional domain (cubature), and, furthermore, can deal with general, empirical basis functions, the qualifier empirical meaning “derived from computational experiments” [39]—hence the employed appellation Empirical Cubature Method.

4.4. Discrete formulation

In formulating the minimization problem (29), it was tacitly assumed that a continuous description of the integrand is available. However, in common finite element implementations, the value of the integrand is only calculated at the integration points of each finite element. For practical reasons, thus, it proves convenient to retrace the analysis described in the foregoing from a “discrete” perspective, i.e., treating the integrand as a spatially discrete variable.

The most immediate implication of the integrand being defined only at the integration points of the mesh is that the integral we wish to approximate can no longer be regarded as a linear operator that maps continuous functions into \mathbb{R} ; rather, it should be viewed as an operator that maps vectors of \mathbb{R}^M (recall that M is the number of FE integration points) to \mathbb{R} . The matrix representation of this operator can be inferred from the expression for the approximated integral of f_I as follows:

$$\begin{aligned} F_I &= \int_{\Omega} f_I \, d\Omega \approx \sum_{g=1}^M W_g f_I(\mathbf{x}_g) = \sum_{g=1}^M \sqrt{W_g} (\sqrt{W_g} f_I(\mathbf{x}_g)) \\ &= \sqrt{\mathbf{W}}^T \mathcal{F}_I, \quad I = 1, 2, \dots, n \end{aligned} \quad (45)$$

where $\sqrt{\mathbf{W}} \in \mathbb{R}^M$ is defined by

$$\sqrt{\mathbf{W}} := \begin{bmatrix} \sqrt{W_1} & \sqrt{W_2} & \dots & \sqrt{W_M} \end{bmatrix}^T, \quad (46)$$

and $\mathcal{F}_I \in \mathbb{R}^M$ is formed by gathering in a single column vector the values of the integrand at all FE points (multiplied by the square root of each finite element integration weight)

$$\mathcal{F}_I := \begin{bmatrix} \sqrt{W_1} f_I(\mathbf{x}_1) \\ \sqrt{W_2} f_I(\mathbf{x}_2) \\ \vdots \\ \sqrt{W_M} f_I(\mathbf{x}_M) \end{bmatrix}. \quad (47)$$

It readily follows from Eq. (45) that $\sqrt{\mathbf{W}}^T$ is but the *matrix representation* of the integral operator when the domain space \mathbb{R}^M is endowed with the standard scalar product.

Remark 4.2. Notice that the definition of operator $\sqrt{\mathbf{W}}^T$ tacitly presupposes that all FE weights are positive. In view of this apparent limitation of the method (it excludes FE integration rules with negative weights), it may be wondered why not to define the operator by putting all the weights in the vector \mathcal{F}_I (that is, $(\mathcal{F}_I)_g = W_g f_I(\mathbf{x}_g)$), or in the operator itself (so that $(\mathcal{F}_I)_g = f_I(\mathbf{x}_g)$). The reason is that the discrete formulation (45) is, strictly speaking, the only one consistent with its continuous counterpart:

$$\|f_I\|_{L_2(\Omega)}^2 = \int_{\Omega} f_I f_I \, d\Omega \approx \sum_{g=1}^M W_g f_I(\mathbf{x}_g) f_I(\mathbf{x}_g) = \|\mathcal{F}_I\|^2 \quad (48)$$

i.e., the euclidean norm of \mathcal{F}_I is the actual approximation of the L_2 norm of $f_I(\mathbf{x})$. Nevertheless, the procedure explained in the following discussion can be also applied when the other two (less restrictive) definitions of the discrete integral operator are adopted (in fact, in the event of meshes with fairly uniform element size, the three alternatives should yield similar results).

4.4.1. Basis matrices

Having defined the discrete version of the integral operator, the next step consists in determining the basis matrix for the nonlinear term \mathcal{F}_I ($I = 1, 2, \dots, n$). According to the previously described expanded basis approach, the basis matrix for \mathcal{F}_I is to be constructed as the union of a basis matrix for the range of $\sqrt{\mathbf{W}}^T$ (which is simply the space spanned by $\sqrt{\mathbf{W}}$) and a basis matrix, $\mathbf{A} \in \mathbb{R}^{M \times p}$, for the projection of \mathcal{F}_I onto the kernel of $\sqrt{\mathbf{W}}^T$, i.e:

$$\text{Expanded basis matrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_p & \sqrt{\mathbf{W}} \end{bmatrix}. \quad (49)$$

The starting point for computing \mathbf{A} is the snapshot matrix of \mathcal{F}_I^j for all components $I = 1, 2, \dots, n$ and all training configurations $j = 1, 2, \dots, P$:

$$\mathbf{X}_{\mathcal{F}} = \begin{bmatrix} \mathcal{F}_1^1 & \dots & \mathcal{F}_n^1 & \mathcal{F}_1^2 & \dots & \mathcal{F}_n^2 & \mathcal{F}_1^P & \dots & \mathcal{F}_n^P \end{bmatrix}. \quad (50)$$

The projection of each column of $\mathbf{X}_{\mathcal{F}}$ onto $\mathcal{N}(\sqrt{\mathbf{W}}^T)$ can be calculated by subtracting its orthogonal projection onto $\mathcal{R}(\sqrt{\mathbf{W}}^T)$; this operation yields

$$\hat{\mathcal{F}}_I^j := \mathcal{F}_I^j - \frac{\sqrt{\mathbf{W}}}{\|\sqrt{\mathbf{W}}\|} \left(\frac{\sqrt{\mathbf{W}}^T}{\|\sqrt{\mathbf{W}}\|} \mathcal{F}_I^j \right) = \begin{bmatrix} \sqrt{W_1}(f_I^j(\mathbf{x}_1) - F_I^j/V) \\ \sqrt{W_2}(f_I^j(\mathbf{x}_2) - F_I^j/V) \\ \vdots \\ \sqrt{W_M}(f_I^j(\mathbf{x}_M) - F_I^j/V) \end{bmatrix}, \quad (51)$$

which is essentially the same expression obtained in the continuous case (40) (the integrand at a given point minus the volume average), but with each entry multiplied by the square root of the corresponding FE integration weight. The required basis matrix can be finally determined by collecting all these vectors in a single matrix $\hat{\mathbf{X}}_{\mathcal{F}} \in \mathbb{R}^{M \times nP}$

$$\hat{\mathbf{X}}_{\mathcal{F}} = \begin{bmatrix} \hat{\mathcal{F}}_1^1 & \dots & \hat{\mathcal{F}}_n^1 & \hat{\mathcal{F}}_1^2 & \dots & \hat{\mathcal{F}}_n^2 & \hat{\mathcal{F}}_1^P & \dots & \hat{\mathcal{F}}_n^P \end{bmatrix} \quad (52)$$

and then applying a dimensionality reduction technique such as the SVD⁸ to obtain an approximated basis matrix of rank p for the column space of $\hat{\mathbf{X}}_{\mathcal{F}}$, i.e.:

$$\hat{\mathbf{X}}_{\mathcal{F}} \approx \mathbf{A} \mathbf{\Sigma} \mathbf{V}_A^T \quad (53)$$

⁸ It is interesting to note that, since each row of $\hat{\mathbf{X}}_{\mathcal{F}}$ is multiplied by the square root of the corresponding finite element weight, the minimization problem underlying the SVD is actually posed in the norm defined by the diagonal matrix of finite element weights (which, in turn, as we saw earlier, is the discrete counterpart of the L_2 norm).

where Σ_A and V_A^T are the matrices of singular values and right singular vectors, respectively, associated to the selected, dominant left singular vectors⁹ $A \in \mathbb{R}^{M \times p}$.

Remark 4.3. The standard SVD is a RAM-intensive operation, and, consequently, manipulating relatively large snapshot matrices may easily exhaust the memory capabilities of the computer at hand. To overcome this difficulty, we propose to adopt two complementary strategies. Firstly, in problems in which the geometry is input-parameter independent, the nonlinearity between the internal forces and displacements is concentrated on the stresses and, therefore, one may obtain the basis matrix for the internal forces from a basis matrix for the stresses themselves. This strategy is specially advantageous for relatively high number of reduced displacement modes (n), as the size of the snapshot stress matrix is independent of this reduced dimension.

For high number of training configurations, however, the above strategy may prove insufficient to lower the offline computational requirements to affordable levels. For such cases, we have devised a partitioned version of the SVD that precludes the necessity of manipulating the whole matrix, hence significantly diminishing memory requirements.

To preserve the continuity of the presentation, the detailed description of these two complementary methods is relegated to [Appendices A](#) and [B](#), respectively.

4.4.2. Minimization problem

The statement of the discrete minimization problem that allows one to determine the optimized location of integration points and corresponding weights is quite similar to the continuous problem (23): find $\alpha \in \mathbb{R}_+^m$ and $\mathbf{z} \in \mathbb{N}^m \subset \{1, 2, \dots, M\}$ such that

$$(\alpha, \mathbf{z}) = \arg \min_{\alpha \geq 0, \mathbf{z}} \|\mathbf{J}_{\mathbf{z}} \bar{\alpha} - \mathbf{b}\|^2. \quad (54)$$

Note that, in this case, the goal is to choose a set of m points *among* the M FE integration points of the underlying finite element mesh $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ —that is, we have now a *combinatorial optimization problem*. Hereafter, the g th optimal point will be denoted by $\bar{\mathbf{x}}_g := \mathbf{x}_{\mathbf{z}_g}$.

Following the same logic that led to Eq. (42), $\mathbf{J}_{\mathbf{z}} \in \mathbb{R}^{p+1 \times m}$ and $\mathbf{b} \in \mathbb{R}^{p+1}$ can be decomposed into the following block matrices

$$\mathbf{J}_{\mathbf{z}} = \begin{bmatrix} \mathbf{A}_{\mathbf{z}} & \sqrt{\mathbf{W}_{\mathbf{z}}} \end{bmatrix}^T, \quad \mathbf{b} = \begin{bmatrix} \mathbf{0}^T & V \end{bmatrix}^T \quad (55)$$

($\mathbf{A}_{\mathbf{z}}$ and $\sqrt{\mathbf{W}_{\mathbf{z}}}$ denotes the block matrices of \mathbf{A} and $\sqrt{\mathbf{W}}$, respectively, formed by the rows corresponding to the indices of the selected points $\mathbf{z} \subset \{1, 2, \dots, M\}$). With this decomposition, the objective function is expressible as

$$\|\mathbf{J}_{\mathbf{z}} \alpha - \mathbf{b}\|^2 = \|\mathbf{A}_{\mathbf{z}}^T \alpha\|^2 + \|\sqrt{\mathbf{W}_{\mathbf{z}}} \alpha - V\|^2. \quad (56)$$

Introduction of the preceding expression into problem (54) leads to the following minimization problem: find $\alpha \in \mathbb{R}_+^m$ and $\mathbf{z} \in \mathbb{N}^m \subseteq \{1, 2, \dots, M\}$ such that

$$(\alpha, \mathbf{z}) = \arg \min_{\alpha \geq 0, \mathbf{z}} \left(\|\mathbf{A}_{\mathbf{z}}^T \bar{\alpha}\|^2 + \left(\sum_{g=1}^m \sqrt{W_{\mathbf{z}_g}} \bar{\alpha}_g - V \right)^2 \right). \quad (57)$$

After solving for α , the sought-after weights can be computed as

$$\omega_g := \sqrt{W_{\mathbf{z}_g}} \alpha_g, \quad g = 1, 2, \dots, m. \quad (58)$$

Observe that, as in the continuous case, the condition that the sum of weights should be as equal as possible to the volume of the domain appears explicitly in the objective function (57).

⁹ Alternatively, instead of directly using as basis matrix the left singular vectors, one may use as basis matrix $\mathbf{A} \leftarrow \mathbf{A} \Sigma_A$. This is equivalent to define the objective function in (57) in terms of the norm defined by such singular values.

4.5. Solution of the optimization problem

We turn now to the solution of the *combinatorial* optimization problem (54). In the field of combinatorial optimization, this type of problem is termed a *hard regressor selection* [40]: we are given a matrix $\mathbf{J} \in \mathbb{R}^{p+1 \times M}$, whose columns are potential regressors, and a vector $\mathbf{b} \in \mathbb{R}^{p+1}$ is to be fit by a linear combination of $m < M$ columns of \mathbf{J} . The problem is to choose the subset of m columns to be used as well as the associated coefficients. One straightforward approach would be to check every possible combination, considering that, for a fixed set of columns \mathbf{z} , the optimal $\boldsymbol{\alpha}$ can be found by solving a nonnegative least-squares problem—using, for instance, the active set algorithm developed by Lawson and Hanson [41]. In principle, this should be done for each of the $\binom{M}{m}$ feasible combinations. Needless to say, since $M \gg m$ (for otherwise the whole reduction procedure would prove pointless), this brute force approach is not viable and recourse to *heuristic* methods, able to at least determine *sub-optimal solutions*, is to be made.

4.5.1. Greedy selection method

The heuristic method employed in the present work is described in the flowchart of [Box IV](#). This method is based on the *greedy* algorithm put forward by An and co-workers in their seminal paper [19]. Two features distinguish our algorithm from that in Ref. [19], namely: (1) the definition of the matrices \mathbf{J} and \mathbf{b} appearing in the objective function (\mathbf{J} is a row-wise orthogonal matrix and all entries of \mathbf{b} are zero but the last one, which is equal to the volume of the domain); and (2) the unrestricted least-squares appearing in Step 3. The reason why $\boldsymbol{\alpha}$ is tentatively computed by an unrestricted least-squares (LS) fit in Step 3 is that we have empirically observed that, in almost all iterations, this operation furnishes vectors $\boldsymbol{\alpha}$ with positive entries, hence precluding the need to solve the expensive nonnegative least-squares problem of Step 5.

4.5.2. Convergence to absolute minimum

Numerical experience shows that the algorithm in [Box IV](#) invariably drives the objective function $\|\mathbf{J}_\mathbf{z}\boldsymbol{\alpha} - \mathbf{b}\|$ to zero when the number of selected points¹⁰ (i.e. columns) equals the number of rows of \mathbf{J} , that is, when

$$m_+ = p + 1. \quad (59)$$

This convergence property can be attributed to one of the distinguishing feature of the proposed cubature scheme, namely, that \mathbf{J} is the transpose of an orthogonal basis matrix, and hence, it has full rank. When $m_+ = p + 1$, the block matrix $\mathbf{J}_\mathbf{z}$ becomes square and, furthermore, since the algorithm selects linearly independent columns, *invertible*. This means that $\boldsymbol{\alpha} = \mathbf{J}_\mathbf{z}^{-1}\mathbf{b}$ and, consequently, $\|\mathbf{J}_\mathbf{z}\boldsymbol{\alpha} - \mathbf{b}\| = 0$.

The linear independence of the selected columns follows from the fact that, at Step 1, the algorithm identifies the new index \mathbf{z}_k as that whose associated column is the *most positively parallel* to the current residual, for this is the column that will reduce the residual norm the most. This necessarily implies that $\mathbf{J}_{\mathbf{z}_k}$ must be linearly independent of the previously selected columns $\{\mathbf{J}_{\mathbf{z}_1}, \mathbf{J}_{\mathbf{z}_2}, \dots, \mathbf{J}_{\mathbf{z}_{k-1}}\}$ —a linearly dependent column carries redundant information and would not, thus, contribute to lower the residual.

5. Hyperreduced-order model

For completeness, the *offline steps* required for determining the set of integration points and their associated weights are summarized¹¹ in [Box V](#). Likewise, the statement of the resulting “hyper-reduced” order problem is set forth in [Box VI](#). It should be stressed that the only difference between this problem and the reduced-order problem shown previously in Section 3 (see [Box III](#)) is the evaluation of the integral of the reduced internal forces—now it requires computation of the integrand at solely $m \ll M$ points. As a consequence, the solving effort associated to the constitutive equations also diminishes by a factor M/m , as well as memory requirements for storing the history of internal variables.

¹⁰ Notice that, at Step 6, points whose associated weights are identically zero are removed from \mathbf{z} . Hence, the cardinality of \mathbf{z} at the end of each iteration coincides with the number of points with strictly positive weights (m_+).

¹¹ The reduced-order operators $\Phi_0, \mathbf{M}, \mathbf{M}_0, \mathbf{R}_{ext}$ and $\mathbf{B}_g = \mathbf{B}_g^h \Phi$ ($g = 1, 2, \dots, M$) were already determined in the offline phase of the first reduction stage, see [Box II](#), are, hence, need not be computed again.

DATA: $\mathbf{J} = [\mathbf{A} \quad \sqrt{\mathbf{W}}]^T \in \mathbb{R}^{(p+1) \times M}$, $\mathbf{b} = [\mathbf{0} \quad \mathbf{V}]^T \in \mathbb{R}^{p+1}$, TOL , m

Initializations

- Set of integration points: $\mathbf{z} \leftarrow \emptyset$, Set of candidate points: $\mathbf{y} \leftarrow \{1, 2, \dots, M\}$
- Nonzero components of $\boldsymbol{\alpha}$: $m_+ \leftarrow 0$, number of iterations $k \leftarrow 1$
- Residual vector: $\mathbf{r} \leftarrow \mathbf{b}$

while $\|\mathbf{r}\|/\|\mathbf{b}\| > TOL$ **AND** $m_+ \leq m$ **do**

1. Compute new point i as: $i = \arg \max_{i \in \mathbf{y}} \tilde{\mathbf{J}}_i^T \mathbf{r} / \|\mathbf{r}\|$, where $\tilde{\mathbf{J}}_j := \mathbf{J}_j / \|\mathbf{J}_j\|$
2. Move i from set \mathbf{y} to set \mathbf{z} ($\mathbf{z} \leftarrow \mathbf{z} \cup i$ and $\mathbf{y} \leftarrow \mathbf{y} \setminus i$)
3. Determine $\boldsymbol{\alpha}$ by (*unrestricted least-squares*):

$$\boldsymbol{\alpha} \leftarrow (\mathbf{J}_z^T \mathbf{J}_z)^{-1} \mathbf{J}_z^T \mathbf{b}$$

4. If all entries of $\boldsymbol{\alpha}$ are nonnegative, go to step 7.
5. Determine $\boldsymbol{\alpha}$ by solving the *nonnegative least-squares problem*

$$\boldsymbol{\alpha} \leftarrow \arg \min_{\tilde{\boldsymbol{\alpha}} \geq 0} \|\mathbf{J}_z \tilde{\boldsymbol{\alpha}} - \mathbf{b}\|^2$$

6. Set $\mathbf{z} \leftarrow \mathbf{z} \setminus \mathbf{z}_0$, (where $\mathbf{z}_0 \subset \mathbf{z} \mid \boldsymbol{\alpha}(\mathbf{z}_0) = 0$), $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}(\mathbf{z})$ and $\mathbf{y} \leftarrow \mathbf{y} \cup \mathbf{z}_0$.
7. Update the residual: $\mathbf{r} \leftarrow (\mathbf{b} - \mathbf{J}_z \boldsymbol{\alpha})$.
8. Set $k \leftarrow k + 1$, $m_+ \leftarrow \text{card}(\mathbf{z})$

end

9. Compute the desired **integration weights** as

$$\omega_g = \sqrt{W_{z_g}} \alpha_g, \quad g = 1, 2, \dots, m$$

Box IV. Empirical cubature method (offline stage). Greedy algorithm for computing an optimal set of integration points $\mathbf{z} \subset \{1, 2, \dots, M\}$ and corresponding weights $\omega \in \mathbb{R}_+^m$.

1. Solve the reduced-order problem (without approximation of internal forces, see Box III) for the set of representative input parameters $\{\boldsymbol{\mu}^i\}_{i=1}^P$ (the same employed for the problem presented in Box III).
2. Store the components of the reduced internal forces at all integration points and all training configurations in the snapshot matrix $\mathbf{X}_{\mathcal{F}} \in \mathbb{R}^{M \times nP}$ (see Eq. (50)). Alternatively, one may store the stresses at the integration points, and then calculate a “compressed” $\hat{\mathbf{X}}_{\mathcal{F}} \in \mathbb{R}^{M \times nq}$ by means of formula (A.5) in Appendix A. Likewise, store the FE integration weights (including the Jacobian) in a vector $\mathbf{W} \in \mathbb{R}^M$.
3. Compute the matrix of zero-integral snapshots $\hat{\mathbf{X}}_{\mathcal{F}}$ by applying formula (51) to each column of matrix $\mathbf{X}_{\mathcal{F}}$.
4. Determine an orthogonal basis matrix $\mathbf{A} \in \mathbb{R}^{M \times p}$ for the column space of $\hat{\mathbf{X}}_{\mathcal{F}}$ as the p leading left singular vectors arising from the SVD of $\hat{\mathbf{X}}_{\mathcal{F}}$.
5. Construct the matrix $\mathbf{J} \in \mathbb{R}^{(p+1) \times M}$ and $\mathbf{b} \in \mathbb{R}^{p+1}$ appearing in the cubature optimization problem as $\mathbf{J} = [\mathbf{A} \quad \sqrt{\mathbf{W}}]^T$ and $\mathbf{b} = [\mathbf{0}^T \quad \mathbf{V}]^T$, where $\mathbf{V} = \sum_{i=1}^M W_i$.
6. Determine the set of integration points $\mathbf{z} \in \mathbb{N}^m$ and their associated weights $\omega \in \mathbb{R}_+^m$ by means of the algorithm described in Box IV.

Box V. Offline operations (second reduction stage).

5.1. Reconstruction of displacement, stress and strain fields

In practice, the output of interest in the hyper-reduced-order problem described in Box VI is rarely the vector of reduced displacements \mathbf{d} *per se*, but rather a derived quantity of either the nodal displacements or the stresses at the Gauss points of the finite element mesh (or both). The vector of nodal displacements can be recovered by multiplying the reduced displacement vectors¹² \mathbf{d} by the corresponding basis matrix, i.e., $\mathbf{d}^h \approx \boldsymbol{\Phi} \mathbf{d}$.

On the other hand, stresses (or any other related variable, such as the internal variables $\boldsymbol{\xi}$) at any integration point \mathbf{x}_g ($g = 1, 2, \dots, M$) of the finite element mesh can be, in principle, recovered by locally integrating in time the

¹² Recall that \mathbf{d} corresponds to the unknown nodal displacements. Prescribed displacements are given by the expression $\mathbf{d}_0^h = \boldsymbol{\Phi}_0 \mathbf{d}_0$, where $\boldsymbol{\Phi}_0$ is defined in Eq. (10).

Given $\boldsymbol{\mu} \in \mathcal{D}$, $\mathbf{g}_b^h(\boldsymbol{\mu}) : [0, T] \rightarrow \mathbb{R}^{n_0}$, $\mathbf{u}_0, \mathbf{v}_0 \in \mathbb{R}^n$, $\mathbf{F}_{ext}^h|_c(\boldsymbol{\mu}) : [0, T] \rightarrow \mathbb{R}^{n_f}$, find $\mathbf{d} : [0, T] \rightarrow \mathbb{R}^n$ such that

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{F} = \mathbf{F}_{ext} - \mathbf{M}_0\ddot{\mathbf{d}}_0$$

where

$$\mathbf{F} = \int_{\Omega} \mathbf{f} \, d\Omega \approx \sum_{g=1}^m \omega_g \mathbf{f}(\mathbf{x}_{z_g}; \cdot), \quad (\mathbf{f} = \boldsymbol{\Phi}^T \mathbf{f}^h)$$

and $\mathbf{F}_{ext} = \mathbf{R}_{ext} \mathbf{F}_{ext}^h|_c$, subject to the Dirichlet boundary and initial conditions:

$$\mathbf{d}_0 = \mathbf{g}_b^h, \quad \mathbf{d}(0) = \mathbf{u}_0, \quad \dot{\mathbf{d}}(0) = \mathbf{v}_0,$$

and to the constitutive equations

$$\mathcal{H}(\boldsymbol{\sigma}, \mathbf{d}, \mathbf{d}_0, \boldsymbol{\xi}; \boldsymbol{\mu})|_{\mathbf{x}_{z_g}} = \mathbf{0}, \quad g = 1, 2, \dots, m$$

Box VI. Statement of the hyper-reduced order problem (with approximation of internal forces).

corresponding constitutive equations. In general, these recovery operations are carried out when the simulation is completed, as part of the postprocess – to display, for instance, contour plot of stresses – and therefore do not affect the online cost of the reduced-order simulation.

However, there are certain problems in which the output of interest is a function of the whole stress field and, in addition, it has to be computed online, at the end of each time step. In such cases, an *efficient* recovery of the stress field is a must. One route for efficiently recovering or *reconstructing* the stresses at all Gauss points is via *least-squares fitting* (the so-called Gappy Data reconstruction, introduced by Everson and Sirovich in [8]). Indeed, let $\boldsymbol{\Psi} \in \mathbb{R}^{s \cdot M \times q}$ be a stress basis matrix (calculate by the SVD, for instance),¹³ and let $\hat{\mathbf{S}}_{\mathbf{z}} \in \mathbb{R}^{s \cdot m}$ denote the vector containing the stresses computed by the hyper-reduced order model at the selected integration points \mathbf{z} , that is:

$$\hat{\mathbf{S}}_{\mathbf{z}} = [\boldsymbol{\sigma}^T(\mathbf{x}_{z_1}) \quad \boldsymbol{\sigma}^T(\mathbf{x}_{z_2}) \quad \dots \quad \boldsymbol{\sigma}^T(\mathbf{x}_{z_m})]^T. \quad (60)$$

Least-squares fitting yields the following approximated, global stress vector $\mathbf{S} \in \mathbb{R}^{s \cdot M}$

$$\mathbf{S} \approx \boldsymbol{\Psi} \boldsymbol{\Psi}_{\mathbf{z}}^+ \hat{\mathbf{S}}_{\mathbf{z}} \quad (61)$$

where $\boldsymbol{\Psi}_{\mathbf{z}}^+ = (\boldsymbol{\Psi}_{\mathbf{z}}^T \boldsymbol{\Psi}_{\mathbf{z}})^{-1} \boldsymbol{\Psi}_{\mathbf{z}}^T$ stands for the pseudo-inverse of $\boldsymbol{\Psi}_{\mathbf{z}} \in \mathbb{R}^{s \cdot m \times q}$ ($\boldsymbol{\Psi}_{\mathbf{z}}$ is the matrix formed by the rows of $\boldsymbol{\Psi}$ corresponding to the set of indices $\mathbf{z} \in \mathbb{N}^m \subset \{1, 2, \dots, M\}$).¹⁴

Recovery or reconstruction by least-squares fitting can be also applied to strains. However, to compute a basis matrix (designated by $\boldsymbol{\Upsilon}$) for strains, one need not to store snapshots as in the case of stresses. Rather, a “compressed” snapshot matrix \mathbf{E}_{ε} (featuring $n + n_0$ columns) can be obtained by multiplying the displacement basis matrices $\boldsymbol{\Phi} \in \mathbb{R}^{N \times n}$ and $\boldsymbol{\Phi}_0 \in \mathbb{R}^{N_0 \times n_0}$ by the corresponding FE strain–displacement matrices. This can be done by defining the global (sparse) displacement basis matrix

$$\underline{\boldsymbol{\Phi}} = \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_0 \end{bmatrix} \quad (62)$$

and then computing the rows of \mathbf{E}_{ε} associated to the FE integration point \mathbf{x}_g ($g = 1, 2, \dots, M$) as

$$\mathbf{E}_{\varepsilon}(\mathbf{x}_g) = \mathbf{B}^{(e)}(\mathbf{x}_g) \underline{\boldsymbol{\Phi}}^{(e)}, \quad (63)$$

$\mathbf{B}^{(e)}(\mathbf{x}_g)$ being the FE strain–displacement matrix of element e at integration point \mathbf{x}_g , and $\underline{\boldsymbol{\Phi}}^{(e)}$ the rows of $\underline{\boldsymbol{\Phi}}$ corresponding to the degrees of freedom of element e (e is the index of the finite element containing the integration

¹³ Notice that this stress basis matrix is to be calculated anyway when internal forces are computed by the method described in [Appendix A](#).

¹⁴ For $\boldsymbol{\Psi}_{\mathbf{z}}^+$ to exist, $\boldsymbol{\Psi}_{\mathbf{z}}$ must be full rank, and this, in turn, requires the number of rows of $\boldsymbol{\Psi}_{\mathbf{z}}$ be greater than the number of columns ($m \cdot s \geq q$). In [Appendix A](#), we show that $p \sim n \cdot q$, which, by virtue of the findings of Section 4.5.2, implies that $m \sim n \cdot q$. Thus, it follows that the necessary condition for $\boldsymbol{\Psi}_{\mathbf{z}}^+$ to exist is largely met.

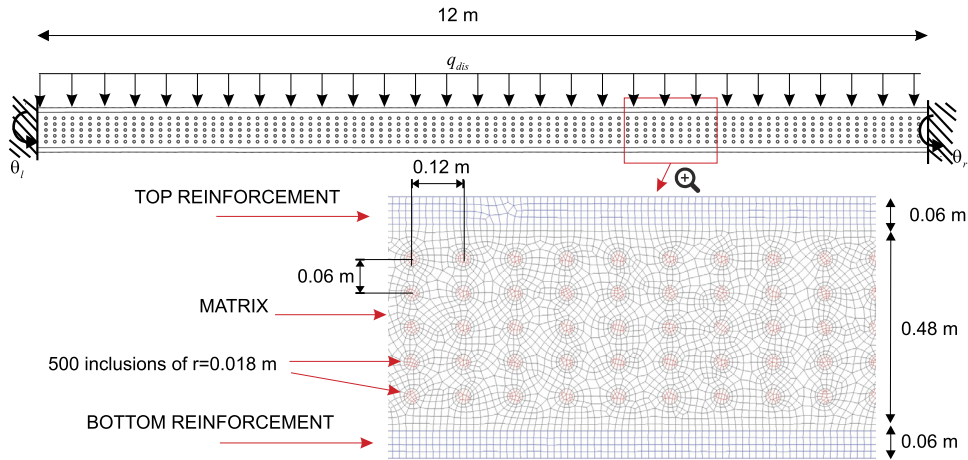


Fig. 1. Geometry of the plate (plane strain), along with the employed FE discretization and boundary conditions.

point \mathbf{x}_g). Once \mathbf{E}_ε is at one's disposal, an orthonormal basis matrix for strains can be determined¹⁵ by an SVD. It is worth noting that, in cases in which strains admit an additive decomposition into elastic and inelastic parts, one may reconstruct with this basis matrix, not only the total strains, but also each of these contributions. This capability is illustrated in the numerical example discussed in the ensuing section (see Fig. 10).

6. Numerical results

In this section, the efficiency of the proposed model-order reduction strategy is assessed in two representative structural examples, namely, a quasistatic (cylindrical) bending problem, on the one hand, and a dynamic, forced vibration problem, on the other hand (both in composite plates undergoing infinitesimal, elasto-plastic deformations).

6.1. Bending of a composite plate

6.1.1. Problem set-up

The composite plate is made of three distinct materials: matrix, reinforcements, and foam. The mechanical behavior of both the matrix and reinforcements materials is modeled by a rate-independent, Von Mises elastoplastic model endowed with a linear, isotropic hardening law (consult Ref. [42] for details on the implementation of this elastoplastic model). The material properties for the matrix material are: Young's modulus $E^m = 70 \cdot 10^3$ MPa, Poisson's ratio $\nu^m = 0.3$, yield stress $\sigma_y^m = 60$ MPa and hardening modulus $H^m = 5$ MPa. For the reinforcement material, on the other hand, these constants take the following values: $E^r = 200 \cdot 10^3$ MPa, $\nu^r = 0.3$, $\sigma_y^r = 110$ MPa and $H^r = 10$ MPa. Lastly, the foam inclusions are assumed to behave elastically, with $E^f = 20$ MPa and $\nu^f = 0.3$.

The finite element mesh can be also seen in Fig. 1. The number of (four-node bilinear) elements is $N_{elem} = 45\,349$, and the number of nodes $N_{node} = 46\,163$. The employed quadrature formula, on the other hand, is the standard 2×2 Gauss rule, the total number of Gauss points amounting thus to $M = 4 N_{elem} = 181\,396$. To overcome incompressibility issues while maintaining the displacement-based formulation presented in the preceding sections, the commonly known as “B-bar” approach is adopted [42]. The constitutive differential equations are integrated in time using the classical (fully implicit) backward-Euler scheme.

6.1.2. First reduction stage

The goal of the hyper-reduced order model (HROM) we wish to develop is to predict the *bending moment on the left edge of the plate for any prescribed rotation on the left and right edges (θ_l and θ_r , respectively) and any transverse, uniformly distributed load (q_{dis})*. The set of input parameters $\boldsymbol{\mu}$ in the problem, thus, can be symbolically

¹⁵ Normally, \mathbf{E}_ε is full rank and can be thereby directly taken as a basis matrix, i.e., $\boldsymbol{\mathcal{T}} = \mathbf{E}_\varepsilon$.

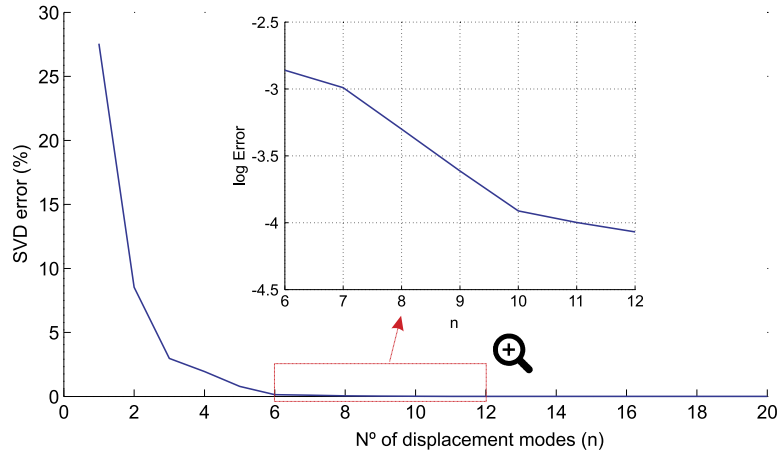


Fig. 2. SVD truncation error versus number of basis vectors employed in the approximation (n). The portion between 6 and 12 modes is shown in magnified form in logarithmic scale.

represented by $\boldsymbol{\mu} = \{\theta_l(t), \theta_r(t), q_{dis}(t)\}$. Notice that the time variable t appears implicitly also in the set of input parameters—due to plastic yielding, the response of the plate at a particular time depends on the *history* of these three parameters.

As explained in the preceding sections, the construction of the desired HROM involves two sequential dimensionality reduction stages. The first reduction stage consists in the creation of the reduced-order model with no approximation of internal forces (henceforth labeled ROM). To arrive at the ROM, we follow the steps outlined in [Box II](#). The first step is to run finite element analysis for representative values of such input parameters (the *training* inputs). We have chosen three sets of such training inputs: rotation of increasing magnitude (linear with time) on the left edge while the right edge remains fixed ($\boldsymbol{\mu}^{tr1} = \{\theta_l^0 t/T, 0, 0\}$), rotation of increasing magnitude on the right edge while the left edge remains fixed ($\boldsymbol{\mu}^{tr2} = \{0, \theta_r^0 t/T, 0\}$), and transverse load of increasing magnitude while the right and the left edges remain fixed ($\boldsymbol{\mu}^{tr3} = \{0, 0, q_{dis}^0 t/T\}$), where $t \in [0, T]$, with $T = 10$ s. The values of the constants appearing in the preceding expression are set to $\theta_l^0 = -\theta_r^0 = 0.025$ rad, and $q_{dis}^0 = -0.85$ MN/m²—these values ensure that the plate is loaded well into the inelastic range and, consequence, plastic hinges develop. The time domain for each input history is discretized into 200 equally spaced steps, resulting in a total number of $P = 3 \cdot 200 = 600$ snapshots.

Three matrices are to be stored in memory and processed by dimensionality reduction in the first reduction stage (Step 2 of [Box II](#)): the matrix of prescribed displacements $\mathbf{X}_d \in \mathbb{R}^{N_0 \times P}$, the matrix of external forces $\mathbf{X}_{fext} \in \mathbb{R}^{N \times P}$, and the matrix of unrestricted displacements $\mathbf{X}_d \in \mathbb{R}^{N \times P}$ (the number of restricted and unrestricted DOFs is $N_0 = 156$ and $N = 2N_{node} - N_0 = 92\,170$, respectively). Spatial variation of boundary conditions can be (exactly) described by just two parameters (θ_l and θ_r), and therefore, $n_0 = \text{rank}(\mathbf{X}_{d0}) = 2$; by the same token, for external forces, $n_f = \text{rank}(\mathbf{X}_{fext}) = 1$. Using the SVD, the corresponding basis matrices $\boldsymbol{\Xi} \in \mathbb{R}^{N_0 \times n_0}$ and $\boldsymbol{\Theta} \in \mathbb{R}^{N \times n_f}$ (see Section 3.2) can be easily obtained; the interpolation indices \mathbf{b} and \mathbf{c} , on the other hand, can be either automatically determined by means of the Empirical Interpolation Method [3,4], or by manually selecting two horizontal degrees of freedom (DOFs) (one on each edge) for interpolation of prescribed displacements, and one vertical DOF on the top boundary for interpolation of external force.

To obtain the basis matrix $\boldsymbol{\Phi}$ for the unrestricted displacements, we follow the SVD-like elastic/inelastic factorization proposed by the authors in Ref. [17]—and sketched, for completeness, in [Appendix C](#). Since the problem is quasi-static in the small strains regime, the number of elastic modes¹⁶ is equal to the spatial dimensionality of the set of input parameters, i.e., $n^{el} = n_0 + n_f = 3$. Therefore, we only have to elucidate how many inelastic modes $n^{in} = n - n^{el}$ can be deemed as dominant or “essential”. To this end, we plot in [Fig. 2](#) the (dimensionless) SVD truncation error estimates defined as $e^* := \|\mathbf{X}_d - \mathbf{X}_d^*(n)\|/\|\mathbf{X}_d\|$, where $\mathbf{X}_d^*(n)$ denotes the SVD approximation of rank n .

It can be appreciated in [Fig. 2](#) that the first three modes – the elastic modes – contains more than 95% of the information, while the remaining 5% only corresponds to pure inelastic modes. With $n = 10$ (7 inelastic modes), the

¹⁶ The number of displacement modes necessary to exactly reproduce the FE results in the elastic range.

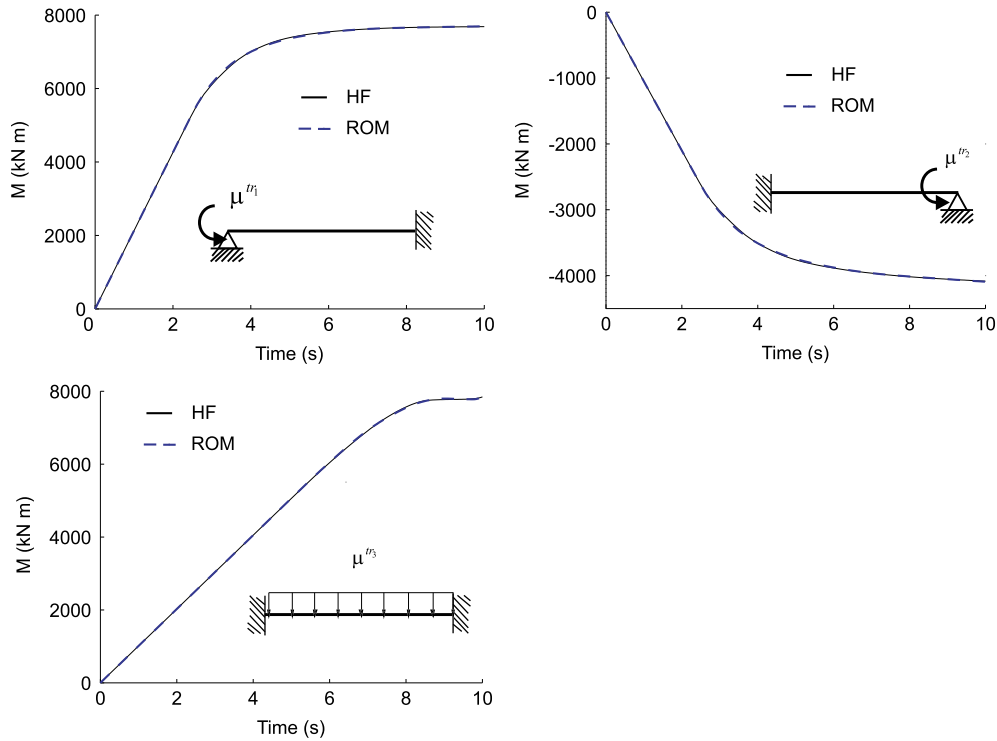


Fig. 3. Bending moment on the left edge versus time for the three training trajectories. Results computed with the FE model and the ROM using $n = 10$ and the full set of FE integration points. The relative L2 errors for the three cases are 0.20%, 0.19% and 0.16%.

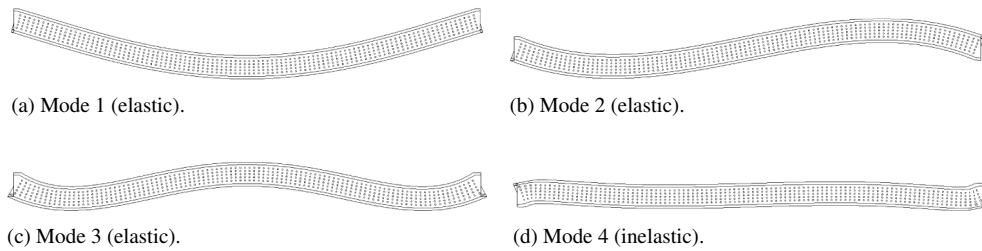


Fig. 4. Deformed shapes corresponding to the first 4 dominant displacement modes ($\{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$).

error level is around 0.01%. In Fig. 3, we plot the output of interest (evolution of bending moment on the left edge) for the three trajectories using the FE model and the ROM constructed with such a number of displacement modes ($n = 10$); as expected, differences between FEM response and the one predicted by the ROM are negligible (below 0.2%). Finally, by way of illustration, Fig. 4 displays the deformed shape corresponding to the first 4 displacement modes ($\Phi_i, i = 1, 2, 3, 4$).

6.1.3. Second reduction stage

In the first reduction stage, discussed in the foregoing, the number of displacement unknowns have been reduced from $N = 92\,170$ to $n = 10$ with almost no loss in accuracy. Yet, this reduction in the number of DOFs only provides modest speedup of around 2. This fact highlights that the actual bottleneck for fast online computation is not the solution of the discrete balance equations, but rather the determination of the stresses, internal forces and stiffness matrices at all the integration points of the underlying finite element mesh (in this case, $M = 181\,396$). The second reduction stage, summarized in Box V, is aimed at choosing among these FE points a reduced set of optimized cubature points (as well as their associated positive weights).

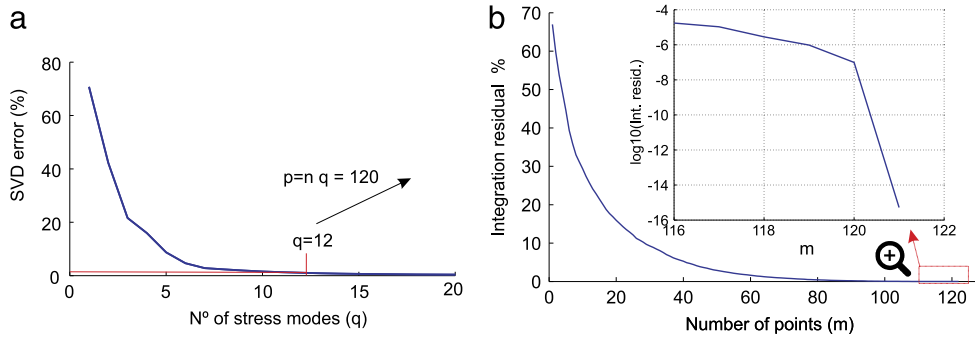


Fig. 5. (a) SVD truncation error for stresses versus number of modes q . (b) Dimensionless residual $\|r\|/V$ versus number of integration points for the case $q = 12$ (hence $p = q \cdot n = 12 \cdot 10 = 120$). The portion between 116 and 122 points is shown in magnified form in logarithmic scale.

We begin by solving the reduced-order equations (Step 1) for the same input parameters employed in the first reduction stage (that is, $\{\mu^{tr1}(t_i), \mu^{tr2}(t_i), \mu^{tr3}(t_i)\}$, $i = 1, 2, \dots, 200$). For computing and processing the reduced internal forces in Step 2, we adopt the stress-based procedure described in [Appendix A](#): the stresses at all FE Gauss points and all time steps for the three trajectories are stored column-wise in a matrix $X_\sigma \in \mathbb{R}^{sM \times P}$ (here $s = 4$); then, the elastic/inelastic factorization (the same employed for displacements) is performed on X_σ , taking as number of elastic stress modes¹⁷ $q^{el} = 3$. The corresponding SVD error graph versus number of stress modes q – the analogous of [Fig. 2](#) – is displayed in [Fig. 5\(a\)](#). It can be gleaned from this Figure that $q = 12$ gives reasonably low stress approximation error (below 1%). Accordingly, we construct the “compressed” internal force matrix X_F (see formula (A.5) in [Appendix A](#)) using $q = 12$ stress modes—hence, X_F features $p = q \cdot n = 12 \cdot 10 = 120$ columns.

Next, we determine the zero-average internal force matrix \hat{X}_F (Step 3 in [Box V](#)), and use the SVD to determine a set of orthogonal basis vectors A for the reduced internal forces (Step 4). With A at our disposal, we construct the matrices J and b appearing in the objective function of the cubature optimization problem (Step 5), and finally solve this problem by means of the greedy algorithm of [Box IV](#) (Step 6).

6.1.4. Empirical cubature

To examine the convergence of this greedy algorithm, we plot in [Fig. 5\(b\)](#) the dimensionless residual $\|r\|/\|b\| = \|r\|/V$ (the one employed as termination criterion in [Box IV](#)) versus the number of points m (for the case $p = n \cdot q = 10 \cdot 12 = 120$). It can be readily seen that the residual decreases monotonically as the number of integration points increases, and at the threshold $m = p + 1 = 121$, it drops sharply to a negligible value ($\sim 10^{-15}$), indicating that, as theoretically anticipated in [Section 4.5.2](#), the algorithm has converged to the absolute minimum.

Concerning the computation time, the partitioned SVD of the stress snapshot matrix took approximately 70 s, while the selection of $m = 121$ points among $M = 181\,396$ FE points was carried out in approximately 45 s (both operations in an serial *Matlab* program, running at 2.9 GHz with 8 GB of RAM and 4 *Intel Core-i7* processors, in Linux). This is as a relatively low computation time when compared with previous cubature optimization methods. When using the original¹⁸ greedy algorithm by An et al. [19], the CPU time for selecting $m = 121$ points rises up to 1014 s (approximately 8 times slower). Difference in performance becomes more pronounced as the number of selected points increases. For $m = 300$ points, for instance, our algorithm needed 233 s (≈ 4 min), while the algorithm in [19] employed 4053 s (1 h and 10 min). As pointed out in [Section 4.5.1](#), the reasons why our method appears to outperform the original scheme of [19] are two: firstly, memory requirements are drastically reduced; and, secondly, and most importantly, our approach avoids in almost all iterations the expensive nonnegative least-squares problem. Specifically, in the selection of $m = 300$ points, our algorithm only needed to solve the nonnegative least squares problem in 5 iterations (of a total of 307 iterations).

¹⁷ In order not to process the whole matrix X_σ , we employ the partitioned SVD strategy, described in [Appendix B](#), for both the elastic and inelastic matrices. In particular, we employ a partition of 10 block matrices with 60 snapshots each.

¹⁸ We implemented the greedy algorithm proposed by [19] in its *raw* form, i.e., without the heuristics aimed at accelerating the algorithm (namely, subset strategy for choosing new candidates, and subset training).

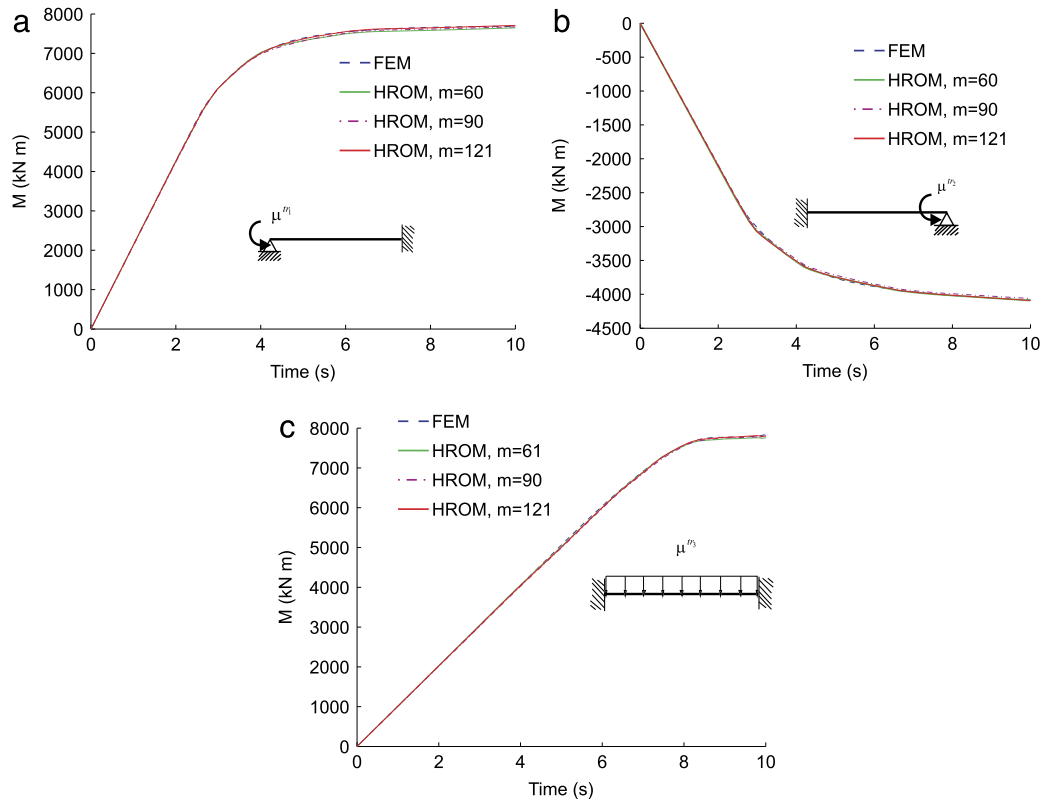


Fig. 6. Bending moment on the left edge versus time for the three training trajectories. Results computed with the FE model ($M = 181\,396$ integration points) and the HROM using $m = 60, 90$ and 121 integration points.

Table 1

Relative L2 error between the predictions of the HROM for the three training trajectories – using $m = 60, 90$ and 121 integration points – and the reference FE model (that employs $M = 181\,396$ integration points).

	Traj. 1	Traj. 2	Traj. 3	Max.
$m = 60$	0.48%	0.30%	0.78%	0.78%
$m = 90$	0.51%	0.68%	0.50%	0.68%
$m = 121$	0.29%	0.31%	0.17%	0.31%

6.1.5. HROM results

Next we study the extent to which the integration error affects the quality of the response predicted by the HROM (in terms of the output of interest). To this end, we analyze in Fig. 6 the evolution of the bending moment on the left edge versus time, for the three training trajectories, using both the finite FE model and the HROM with $q = 12$ stress modes and varying number of integration points. Specifically, we set $m = 60, 90$ and 121 . The relative L2 errors for each case are displayed in Table 1. Inspection of Fig. 6 and Table 1 shows that, remarkably, deviations between FE response and the HROM graphs are practically imperceptible for the three cases. These results also suggest that there is no need to exactly integrate all the internal force modes: gains in accuracy achieved by passing from $m = 60$ to $m = p + 1 = 121$ points are practically negligible (from 0.78% to 0.31%).

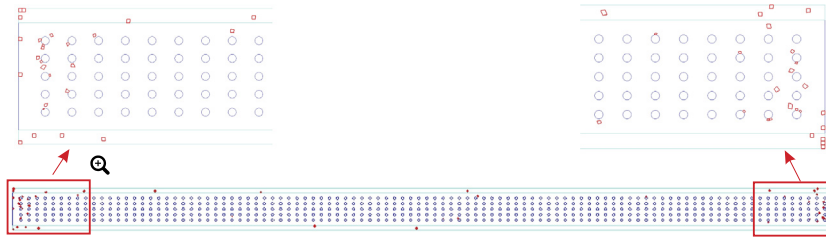


Fig. 7. Location of the $m = 60$ integration points chosen by the greedy algorithm (for $p = 12$ stress modes).

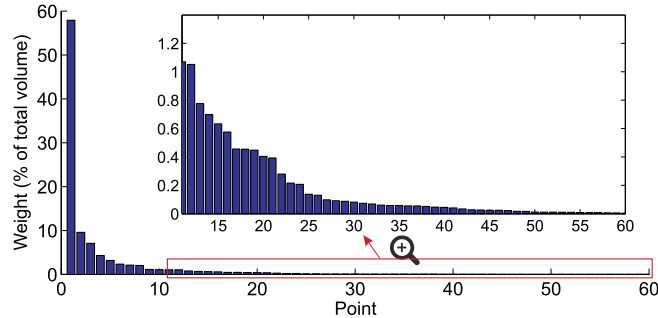


Fig. 8. Weight (in % of the total volume V , and sort in descending order) associated to each integration point (for $m = 60$).

Fig. 7 depicts the position of the elements¹⁹ containing the $m = 60$ points selected by the greedy algorithm. Observe that around 50 points are equally distributed along regions close to the right and left edges, while the remaining points are scattered over the middle portion of the beam. Likewise, the weights (divided by the total volume) associated to these $m = 60$ points are plotted in Fig. 8. Notice that the distribution of the total volume among the chosen points is far from being uniform: the sum of the 10 highest weights amounts to 90% of the total volume, while the remaining weights only contribute with a 9.99% (hence, the error in integrating the volume with these 60 points is only 0.01%).

We pointed out in Section 5.1 that, although the HROM only tracks the evolution of stresses and (plastic and elastic) strains at the reduced set of integration points, one can “reconstruct”, for post-processing purposes, such fields by using least-square fitting (see Eq. (61)). To illustrate this capability, we plot in Figs. 9 and 10 the contour plots of effective stresses and effective plastic strains, respectively, at the end of the first training configuration computed by the FE model and the HROM model (using $q = 12$ stress modes, $n + n_0 = 12$ strain modes and $m = 60$ points). The qualitative resemblance between the HROM and FE patterns is startling in both cases—despite the fact that the number of integration points has been reduced by a factor over 3000. Obviously, upon closer inspection, some “inconsistencies” become apparent. For instance, in Fig. 10(b), it can be appreciated that the HROM predicts that plastic yielding occurs at some of the (elastic) foam inclusions. This is because least-squares fitting is a purely data driven approach and, consequently, the reconstructed variables are not consistent, in general, with the constitutive behavior assumed for each material.

6.1.6. Testing trajectory

We examine now the error incurred by the HROM in predicting the response of the plate for input parameters different from those employed in the “training” process. The plate is subjected to a loading/unloading cycle of prescribed rotation on the right edge in tandem with transverse load, while the left edge remains fixed, i.e., $\mu = \{0, \theta_r^{max} g(t), q_{dis}^{max} g(t)\}$. The constants are set to $\theta_r^{max} = -0.01$ rad and $q_{dis}^{max} = -0.4$ M N/m²; on the other hand, $g = g(t) : [0, T] \rightarrow \mathbb{R}$ is the piecewise constant function shown in Fig. 11(a). The analysis required 300 equally spaced time steps. The plots of bending moment on the left edge versus time for the FE model and the HROM ($m = 60$

¹⁹ It should be stressed here that the selection of a Gauss point within a given finite element does not imply that the rest of Gauss points of the element are included in the reduced set of integration points. Hence, as distinct from the “mesh-sampling” method advocated by Farhart and co-workers [20,21], the HROM proposed here completely ignores the finite element origin of the integration points.

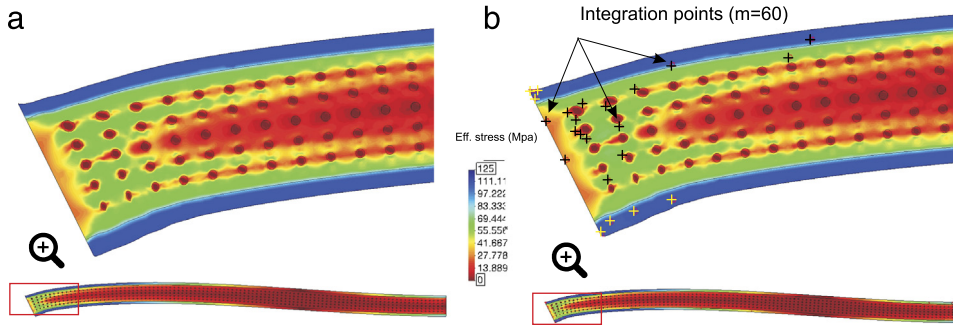


Fig. 9. Contour plot of effective stresses at the end of the first training configuration (deformed shape amplified by a factor of 10). (a) Finite element model ($M = 181\,396$ integrations points). (b) Hyper-reduced order model (“reconstruction” using $q = 12$ stress modes and $m = 60$ integration points).

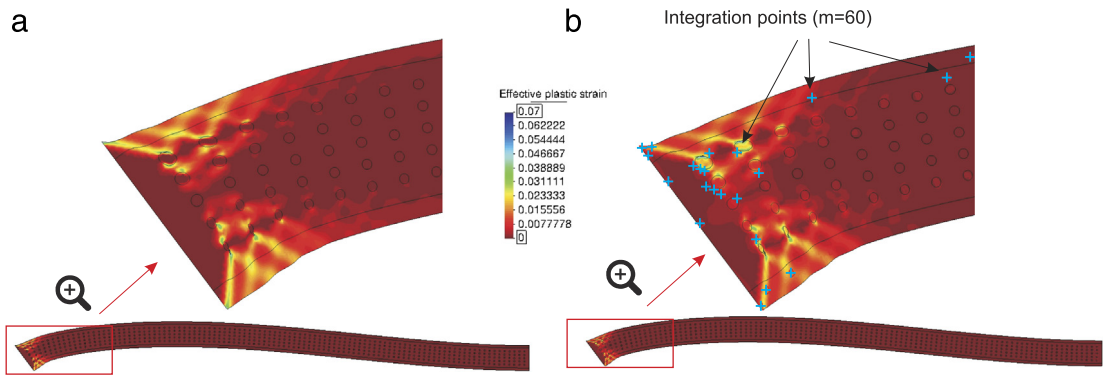


Fig. 10. Contour plot of effective plastic strains at the end of the first training configuration (deformed shape amplified by a factor of 10). (a) Finite element model ($M = 181\,396$ integrations points). (b) Hyperreduced order model (“reconstruction” using $n + n_0 = 10 + 2 = 12$ strain modes and $m = 60$ integration points).

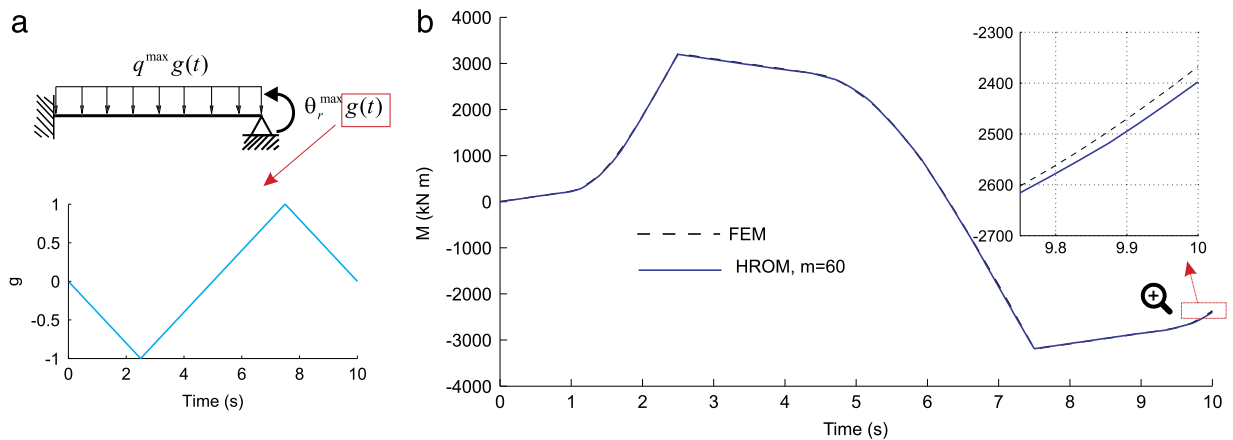


Fig. 11. (a) Input parameters employed to test the HROM. (b) Bending moment on the left edge versus time for these input parameters. Results computed with the FE model and the HROM using $m = 60$ integration points.

integration points) are displayed in Fig. 11(b). Remarkably, the two curves are practically indistinguishable at the scale of plot; the maximum error is observed at the end of the process (residual bending moment, see enlarged region), and it is below 1.5%.

Table 2

Comparison of the dimensions and computation time of the finite element problem and the hyper-reduced order problem, along with the corresponding “compression” ratios.

	FEM	HROM	“Compression” ratio
Number of unrestricted DOFs	$N = 92170$	$n = 10$	9217
Number of restricted DOFs	$N_0 = 156$	$n_0 = 2$	78
Number of parameters defining external forces	$N_f = 776$	$n_f = 1$	776
Number of integration points	$M = 181396$	$m = 60$	3023.2
Computation time	6180 s	2.5 s	2472

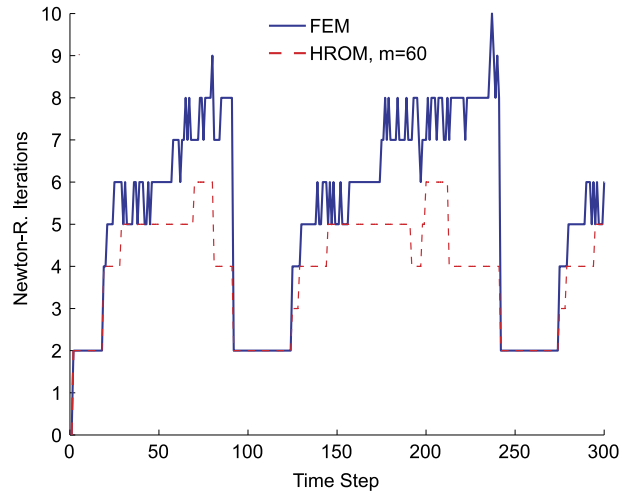


Fig. 12. Number of iterations required for convergence of the accompanying Newton–Raphson algorithm for both the FE model and the HROM (with $m = 60$ points).

6.1.7. Compression ratios and speedup

To summarize, we present in Table 2 the dimensions characterizing both the high-fidelity FE model and the employed HROM, along with the corresponding “compression” or dimensionality reduction factors. Observe that both the number of displacement unknowns and the number of integration points have been reduced by three orders of magnitude. These startling compression ratios are reflected in equally astonishing speedup factors (≈ 2400): the 300 time steps of this example were computed in 2.5 s by the HROM, while the FE model employed 1 h and 43 min (both in a vectorized²⁰ *Matlab* program, running at 2.9 GHz with 8 GB of RAM and 4 *Intel Core-i7* processors, in Linux).

Lastly, to assess the robustness of the HROM, we show in Fig. 12 the number of iterations required for convergence of the accompanying Newton–Raphson algorithm in the case of the FE model and the HROM. Observe that, at all time steps, the number of iterations employed by the HROM is less or equal than in finite element model (for the same convergence tolerance).

6.2. Forced vibration of a composite plate

6.2.1. Problem set-up

We now turn our attention to the development of an HROM for the free vibration of the composite plate shown in Fig. 13. The procedure to arrive at such a HROM is essentially the same as the one explained in the previous example; thereby, in the interest of brevity, some details will be omitted (such as the SVD error analyses).

The plate is made of two materials: matrix and foam. The mechanical behavior of the matrix is modeled by a rate-independent, Von Mises elastoplastic model endowed with a linear, isotropic hardening law ($E^m = 100 \cdot 10^3$ MPa,

²⁰ It should be remarked that the speedup factors are even more spectacular when standard, non-vectorized FE Matlab codes are used; in our case, the speedup for this example rises above 18 000.

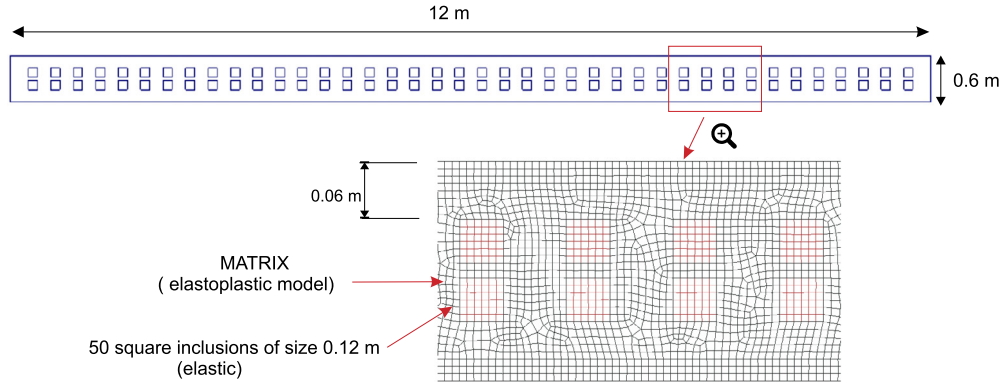


Fig. 13. Geometry of the plate (plane strain), along with the employed FE discretization.

$\nu^m = 0.3$, $\sigma_y^m = 60$ MPa and $H^m = 5$ MPa), while the foam inclusions are assumed to behave elastically ($E^f = 20$ MPa and $\nu^f = 0.3$). The density of the matrix and the foam, on the other hand, are $\rho^m = 7.75 \cdot 10^3$ kg/m³ and $\rho^f = 0.775$ kg/m³, respectively.

The finite element mesh can be also seen in Fig. 13. The number of (four-node bilinear) elements is $N_{elem} = 17\,829$ (hence, $M = 4 \cdot N_{elem} = 71\,316$), and the number of nodes $N_{node} = 18\,460$. The constitutive differential equations are integrated in time using the classical (fully implicit) backward-Euler scheme, while the momentum equation is integrated using a Newmark β -method [43], with parameters $\beta = 1/4$ and $\gamma = 1/2$.

6.2.2. Training process

The goal of the reduced-order model in this problem is to predict the vibration behavior of the plate when the left edge is subjected to a vertical, oscillatory displacement of the form

$$v_l(t) = \sum_{i=1}^Q A_i \sin 2\pi \frac{t}{T_i^v} \quad (64)$$

while the other edge remains free (hence, $\mu = \{v_l(t)\}$). In particular, we are interested in capturing the vibration behavior when the periods of the prescribed displacement T_i^v ($i = 1, 2, \dots, Q$) range between the first and fourth largest *natural* period of the plate: $\bar{T}_1 \leq T_i^v \leq \bar{T}_4$, including the possible development of plastic yielding when $T_i^v \approx \bar{T}_1$ and $T_i^v \approx \bar{T}_2$, that is, when resonance occurs at the two largest natural periods of the structure (an eigenvalue analysis shows that the natural periods of this structure are: $\bar{T}_1 = 0.760$ s, $\bar{T}_2 = 0.125$ s, $\bar{T}_3 = 0.047$ s and $\bar{T}_4 = 0.029$ s).

The training process is made according to the above specifications: we set $Q = 2$, $A_1 = A_2 = 2.1 \cdot 10^{-3}$ m, and $T_1^v = 0.98 \bar{T}_1 = 0.745$ s and $T_2^v = 1.08 \bar{T}_2 = 0.135$ s. The resulting graph of prescribed displacement versus time is displayed in Fig. 14 (blue line).

6.2.3. First reduction stage

We run a finite element analysis in which the time domain is discretized into 6000 equally spaced steps and store in memory the matrix of prescribed displacements $X_{d0} \in \mathbb{R}^{N_0 \times P}$ and the matrix of unrestricted displacements $X_d \in \mathbb{R}^{N \times P}$ (the number of restricted and unrestricted DOFs is $N_0 = 77$ and $N = 2N_{node} - N_0 = 36843$, respectively). Notice that spatial variation of Dirichlet boundary conditions can be described by just one parameters (v_l), and therefore, $n_0 = \text{rank}(X_{d0}) = 1$; by the same token, for external forces, $n_f = 0$.

To obtain the basis matrix Φ for unrestricted displacements, we follow the approach described in Appendix C.1; in this approach, the first n^{vib} columns of Φ are the dominant natural vibration modes of the structure, while the remaining modes are determined by a weighted SVD of the orthogonal complement of the snapshot matrix X_d . Here, we set $n^{vib} = 4$ (to meet of the requirement outlined earlier), and $n^\perp = 6$. These $n = 10$ modes proved to be sufficient for predicting the displacement response of the structure with an error level below 1%. This can be appreciated in

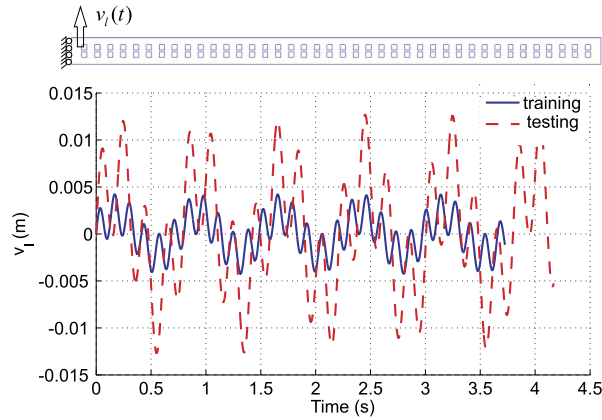


Fig. 14. Prescribed displacements versus time employed for training and testing the HROM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

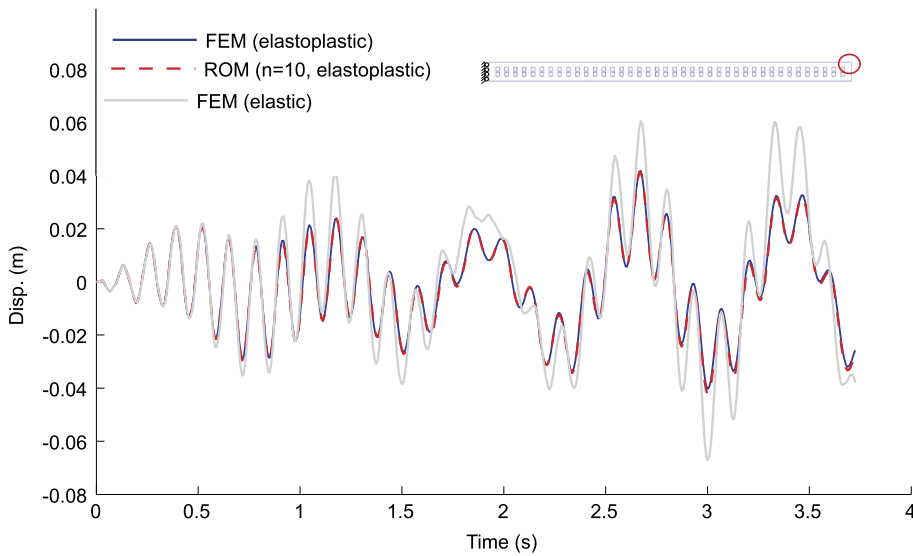


Fig. 15. Displacement of the right, top corner node versus time computed with the FE model and the ROM (using n^{vib} vibration modes and $n^\perp = 6$ SVD modes).

Fig. 15, where we show the vertical displacement of the top, rightmost corner node versus time computed by the FE model and the ROM with these $n = 10$ modes. To illustrate the above mentioned “attenuation” effect due to plastic yielding, we show also in this Figure the displacement computed by the FE model had the matrix material been entirely elastic.

6.2.4. Second reduction stage

For the second reduction stage, we adopt (as in the quasi-static example) the stress-based procedure outlined in Appendix A: the stresses at all FE Gauss points and all time steps are stored column-wise in a snapshot matrix, and then, the SVD-like elastic/inelastic factorization is performed on such a matrix. An error analysis revealed that, by using 4 elastic modes and 6 inelastic modes, the truncation error is below 1%. Accordingly, we construct the required internal force matrix $\hat{\mathbf{X}}_{\mathcal{F}}$ using these modes (hence, $\hat{\mathbf{X}}_{\mathcal{F}}$ has $p = q \cdot n = 10 \cdot 10 = 100$ columns). With $\hat{\mathbf{X}}_{\mathcal{F}}$ at our disposal, we construct the matrices \mathbf{J} and \mathbf{b} appearing in the objective function of the cubature optimization problem, and finally solve such a problem by means of the greedy algorithm of Box IV.

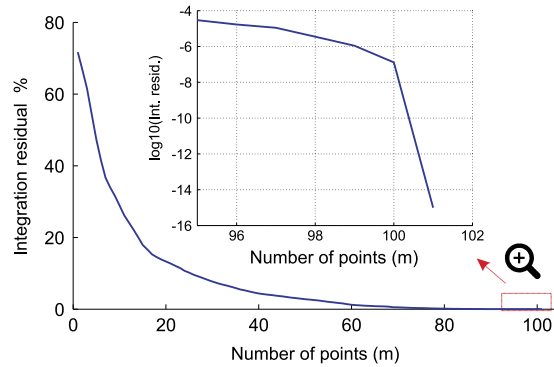


Fig. 16. (a) Dimensionless residual $\|r\|/V$ versus number of integration points for the case $q = 10$ (hence $p = q \cdot n = 10 \cdot 10 = 100$ internal force modes). The portion between 95 and 102 points is shown in magnified form in logarithmic scale.

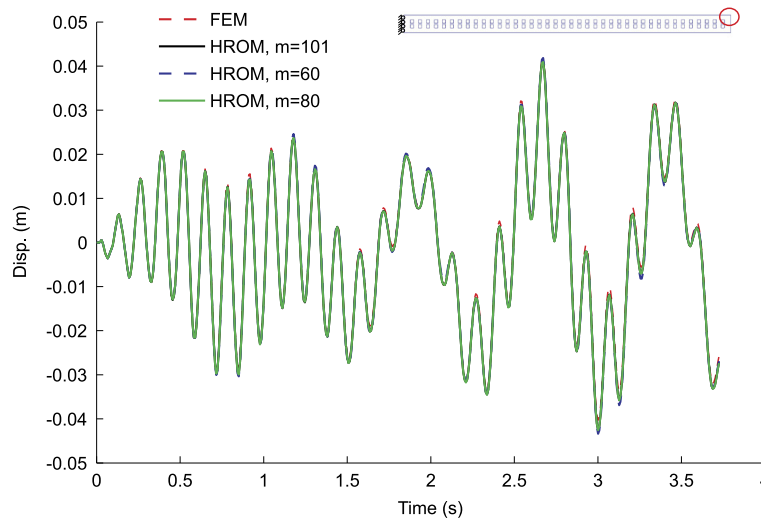


Fig. 17. Displacement of the right, top corner node versus time computed with the FE model and the HROM (using $q = 10$ stress modes and $m = 60, 80$ and 101 integration points).

6.2.5. Empirical cubature

Fig. 16(b) contains the graph of the dimensionless residual $\|r\|/\|b\| = \|r\|/V$ versus the number of points m (for the case $p = n \cdot q = 100$). As expected, the algorithm converges to the absolute minimum (zero integration error) at $m = p + 1 = 101$ points. To assess a posteriori integration errors, we compare in Fig. 17 the plots of displacement of the top, right node versus time for the FE model and the HROM with varying number of integration points ($m = 60, 80$ and 101). Inspection of these plots shows that the level of accuracy at $m = 60$ (residual error around 1%) and $m = 101$ (residual error around 10^{-15}) are quite similar. This observation reinforces the conclusion made in the previous example that there is no need to exactly integrate all the internal force modes appearing in the objective function. On the other hand, Fig. 18(a) shows the location of the elements containing the selected points (for $m = 60$). Observe that around 60% of such points are located in the region near the left edge, wherein plastic yielding is more likely to occur due to resonance-induced bending. Likewise, the value of the associated weights (as a percentage of the total volume) is displayed in Fig. 18(b). The sum of these $m = 60$ weights furnishes a volume only $10^{-5}\%$ below the total volume.

6.2.6. Testing trajectory

It only remains to assess the capability of the HROM to predict the vibrational behavior of the plate under conditions different from those used in the “calibration” (training) process. To this end, we set in Eq. (64) $Q = 2$, $A_1 = A_2 = 6.33 \cdot 10^{-3}$ m, and in order to induce resonance, $T_1^v = 1.02 \bar{T}_1 = 0.758$ s (2% above the first natural

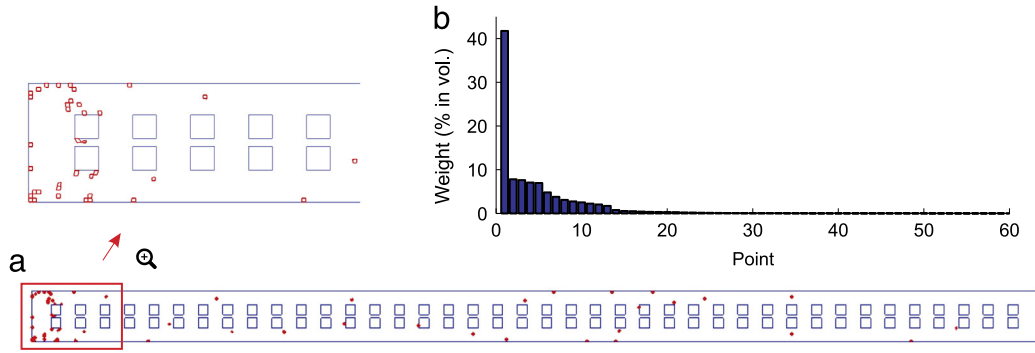


Fig. 18. (a) Location of the $m = 60$ integration points chosen by the greedy algorithm (for $p = 10$ stress modes). (b) Weights (sort in descending order) in % of total volume.

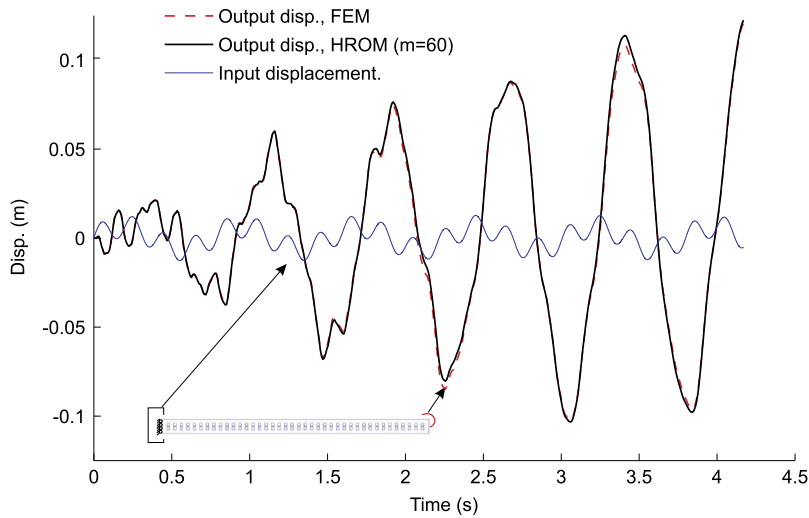


Fig. 19. Displacement of the right, top corner node versus time computed with the FE model and the HROM with $m = 60$ points. The input displacement (prescribed on the left edge) is also shown.

period). The other constant, T_2^v , is set 60% above the second natural period ($T_2^v = 1.6 \bar{T}_2 = 0.20$ s). The resulting prescribed displacement versus time graph is shown in Fig. 14 (red, dotted curve).

In Fig. 19, we show the vertical displacement of the top, rightmost corner node versus time computed by the FE model and the HROM (with $n = 10$ displacement modes and $m = 60$ integration points). To illustrate the resonance phenomenon, this graph is accompanied by the plot of the input displacement versus time (displacement prescribed at the left edge). Discrepancies between the predictions of the high-fidelity, FE model ($M = 71\,316$ integration points) and the hyperreduced-order model ($m = 60$ integration points) are barely perceptible until time $t \approx 1.75$ s; thereafter, small drifts (below 5%) are detected at the peaks of the curve. Remarkably, at the end of the simulation, the HROM error is below 1.5%. The remarkable accuracy of the HROM can be also appreciated in term of stresses, in Fig. 20, where we display the contour plots of effective stresses (at the end of the simulation) obtained by the FEM and the HROM (using $q = 10$ stress modes and $m = 60$ points).

7. Concluding remarks

- **Robustness.** One of the most attractive features of the proposed hyper-reduced order model – and in general, of all cubature-based ROMs – is that it preserves the spectral properties of the Jacobian matrix of the finite element motion equations. This has been corroborated by the examples shown in the preceding section. The materials of the studied composite beams obey small strains, elastoplastic constitutive equations endowed with strain hardening,

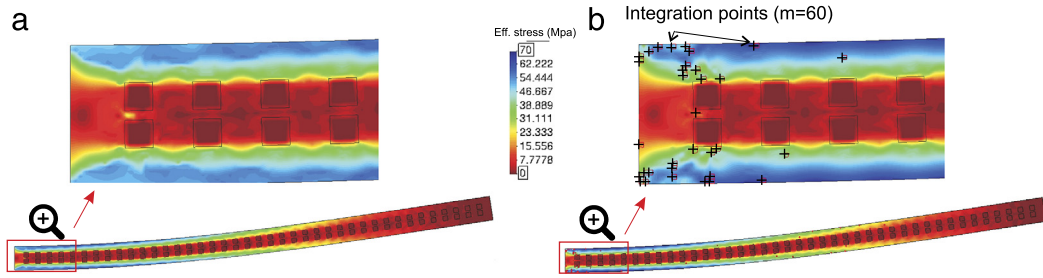


Fig. 20. Contour plot of effective stresses at the end of the simulation. (a) Finite element model ($M = 71\,316$ integration points). (b) Hyperreduced order model (“reconstruction” using $q = 10$ stress modes and $m = 60$ integration points). Deformed shaped amplified 10 times in both cases.

and therefore, the finite element and the reduced order stiffness matrices at all Gauss points are symmetric and positive definite. Since the stiffness matrices in the HROM are positive combination of these matrices, it follows that the HROM matrices are also symmetric and positive definite. As the graphs shown in Fig. 12 demonstrate, this translates into equally fast or even faster convergence rates of the accompanying solution algorithm (Newton–Raphson).

- **Number of integration points.** We have theoretically demonstrated and numerically confirmed that the number of points needed to *exactly* integrate p internal force modes (plus the volume) is $m = p + 1$. It turn, it is shown in Appendix A that the number of internal force modes is equal to the product of numbers of displacement and stress modes: $p = nq$. Thus, it may be concluded that

$$n \leq m \leq nq + 1 \quad (65)$$

(the lower bound n is dictated by well-posedness considerations, see inequality (32)). Furthermore, in most problems (specially those in which the input parameters are variations of boundary conditions), the number of stress modes is of the same order than the number of displacement modes ($q = \mathcal{O}(n)$); thus, we can say that:

$$n \leq m \leq \mathcal{O}(n^2). \quad (66)$$

For instance, in the quasistatic example of Section 6.1, $n = 10$ and $q = 12$, while in the vibration problem of Section 6.2, $n = q = 10$; this amounts to $m = 121$ and $m = 101$ points, respectively, for exactly integrating the corresponding internal force modes.

Nevertheless, the a posteriori accuracy assessments for varying number of integration points carried out for the two examples (Figs. 6 and 17) suggest that *exactly* integrating the chosen p modes amount to “overkill”: in both cases cases, using around 50% and 60% of the threshold $p + 1$ was sufficient to deliver reasonably accurate predictions of the pertinent outputs of interest.

- **Computer implementation.** Three distinct simulation codes are involved in the overall model reduction process, namely, (1) the finite element code, employed for generating displacement snapshots (see Box I); (2) the intermediate reduced-order code, employed for generating the reduced internal force snapshots (see Box III); (3) and the final hyperreduced-order code, Box VI, which is the one used in the “online” computations. It is noteworthy that the three underlying problems are nothing but particular cases of Galerkin approximation method, the basic differences being the basis functions employed for the approximation and the integration rule for the evaluation of internal forces. Indeed, the finite element model uses classical shape functions with local support, while both the ROM and the HROM seek the solution in spaces spanned by Ritz (globally supported) basis functions. Likewise, internal forces (and therefore stiffness matrices) are evaluated by elementwise Gauss rules in the FEM and the ROM, whereas, in the HROM, this operation is carried by a tailored cubature scheme—the proposed Empirical Cubature Method.
- **Empirical Cubature Method.** A distinguishing feature of the proposed Empirical Cubature Method with respect to similar cubature schemes is that the weights at almost all iterations of the greedy selection algorithm are calculated with a standard, unconstrained least-squares—in fact, the nonnegative least squares problem of step 5, see Box IV, is included to filter out small negative weights caused by roundoff errors. We have not given a formal proof of this salient feature – it was discovered by “serendipity” –, but numerical experiments seem to suggest that it may be attributed to the fact that basis vectors in the objection function are mutually orthogonal, and furthermore, their volumetric averages are zero. Research in this aspect of the method is currently in progress and will be reported in a forthcoming publication.

Acknowledgments

The research leading to these results has received funding from, on the one hand, the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 320815, Advanced Grant Project COMP-DES-MAT, and, on the other hand, the Spanish Ministry of Economy and Competitiveness through the National Research Plan 2014: MAT2014-60919-R.

Appendix A. Basis matrix for internal forces

A key step in the development of the proposed reduced-order integration scheme (see Section 4.4) is the determination of an orthogonal basis matrix $\mathbf{A} \in \mathbb{R}^{M \times p}$ for the column space of matrix $\hat{\mathbf{X}}_{\mathcal{F}}$. The direct route to determine this basis matrix is to store the reduced internal forces at all FE integration points and all training configuration in the snapshot matrix $\mathbf{X}_{\mathcal{F}} \in \mathbb{R}^{nM \times P}$, then use Eq. (51) to transform $\mathbf{X}_{\mathcal{F}}$ into $\hat{\mathbf{X}}_{\mathcal{F}}$, and finally apply the SVD to $\hat{\mathbf{X}}_{\mathcal{F}}$ ($\hat{\mathbf{X}}_{\mathcal{F}} \approx \mathbf{A} \mathbf{\Sigma}_A \mathbf{V}_A^T$).

Alternatively, in problems in which the geometry is input-parameter independent, one may obtain the basis matrix for the internal forces from a basis matrix for the stresses. Indeed, the reduced internal force at a given FE integration point \mathbf{x}_g is given by

$$f_I(\mathbf{x}_g; \boldsymbol{\mu}) = \mathbf{B}_I^T(\mathbf{x}_g) \boldsymbol{\sigma}(\mathbf{x}_g, \boldsymbol{\mu}). \quad (\text{A.1})$$

Here, $\mathbf{B}_I \in \mathbb{R}^{s \times 1}$ denotes the reduced strain–displacement matrix for the displacement mode $\boldsymbol{\Phi}_I$, and $\boldsymbol{\sigma} \in \mathbb{R}^s$ the Cauchy stress vector ($s = 4$ or 6 for 2D or 3D problems, respectively). In turn, the expression for \mathbf{B}_I reads

$$\mathbf{B}_I(\mathbf{x}_g) = \mathbf{B}^{(e)}(\mathbf{x}_g) \boldsymbol{\Phi}_I^{(e)} \quad (\text{A.2})$$

$\mathbf{B}^{(e)}(\mathbf{x}_g)$ being the FE strain–displacement matrix of element e at point \mathbf{x}_g , and $\boldsymbol{\Phi}_I^{(e)}$ the entries of $\boldsymbol{\Phi}_I$ corresponding to the degrees of freedom of element e (e denotes the index of the finite element containing the integration point \mathbf{x}_g). Since the geometry is input-independent, \mathbf{B}_I is also independent²¹ of $\boldsymbol{\mu}$; thus, *it is solely the stresses that depend on the input parameters*. The dimensionality reduction effort can be thereby concentrated on the matrix of stress snapshots, constructed by collecting the stress vectors for all FE integrations points and all training configurations as follows:

$$\mathbf{X}_{\sigma} := [\mathbf{S}^1 \quad \mathbf{S}^2 \quad \dots \quad \mathbf{S}^P], \quad (\text{A.3})$$

where $\mathbf{S}^j \in \mathbb{R}^{sM}$ is formed by stacking the stress vector at all FE integration points in a single column vector:

$$\mathbf{S}^j := \begin{bmatrix} \boldsymbol{\sigma}^j(\mathbf{x}_1) \\ \boldsymbol{\sigma}^j(\mathbf{x}_2) \\ \vdots \\ \boldsymbol{\sigma}^j(\mathbf{x}_M) \end{bmatrix}. \quad (\text{A.4})$$

Suppose that the SVD is applied to \mathbf{X}_{σ} , and let $\boldsymbol{\Psi} \in \mathbb{R}^{sM \times q}$ and $\{\bar{\lambda}_i\}_{i=1}^q$ denote the corresponding left singular vector matrix (of rank q) and their associated singular values, respectively. $\boldsymbol{\Psi}$ (with each column multiplied by its corresponding singular values) can be construed as a “compressed” matrix of stress snapshots, and, accordingly, one may calculate the required internal forces matrix $\mathbf{X}_{\mathcal{F}}$ using this compressed version; a generic column of $\mathbf{X}_{\mathcal{F}}$ calculated this way is given by

$$\mathcal{F}_I^j = \begin{bmatrix} \sqrt{W_1} \mathbf{B}_I^T(\mathbf{x}_1) \bar{\lambda}_j \boldsymbol{\Psi}_j(\mathbf{x}_1) \\ \sqrt{W_2} \mathbf{B}_I^T(\mathbf{x}_2) \bar{\lambda}_j \boldsymbol{\Psi}_j(\mathbf{x}_2) \\ \vdots \\ \sqrt{W_M} \mathbf{B}_I^T(\mathbf{x}_M) \bar{\lambda}_j \boldsymbol{\Psi}_j(\mathbf{x}_M) \end{bmatrix}, \quad I = 1, 2, \dots, n, \quad j = 1, 2, \dots, q. \quad (\text{A.5})$$

²¹ This assertion is only true in small strains. To derive a similar decomposition for large strains problems, one has to replace the Cauchy stress vector $\boldsymbol{\sigma}$ by the first Piola–Kirchhoff stress vector.

Notice that the size of the “compressed” matrix $\mathbf{X}_{\mathcal{F}}$ is $M \times qn$, i.e., the number of columns of $\mathbf{X}_{\mathcal{F}}$ is equal to the product of two reduced dimensions: the number of displacement modes n and the number of stress modes q .

The benefits of using this alternative method based on determining the dominant stress modes are twofold. Firstly, memory requirements are drastically reduced, since one need not allocate additional memory for storing the reduced internal forces at all FE points and for all training configurations. Secondly, as opposed to the snapshot matrix of reduced internal forces, the snapshot matrix of stresses is independent of the reduced dimensions of the problem. As a consequence, for problems in which $n > s$, it proves more advantageous in terms of computational cost to apply the SVD to the stress matrix \mathbf{X}_{σ} rather than to the internal force matrix $\hat{\mathbf{X}}_{\mathcal{F}}$. Besides, in problems in which the output of interest depends on stress quantities, the stress basis matrix Ψ is needed for reconstructing the stress field anyway, and therefore, this stress-based strategy would not involve additional computational effort.

Appendix B. Partitioned SVD

For large snapshot matrices, attempting to directly calculate the Singular Value Decomposition (SVD) may exceed the memory capabilities of the computer at hand. In what follows, we propose an (approximate) method for calculating the SVD that precludes the necessity of manipulating the whole matrix, and, therefore, can help in diminishing memory requirements. The method is based on the observation that, in general, contiguous columns within the snapshot matrix exhibit high degree of linear correlation.

Suppose we wish to *approximate* the truncated SVD of rank n of a matrix $\mathbf{X} \in \mathbb{R}^{N \times P}$ (with $N \geq P$, and $n \ll P$). The expression for the *exact* factorization reads

$$\mathbf{X} = \mathbf{U} \Sigma \mathbf{V}^T + \mathbf{E}, \quad (\text{B.1})$$

where $\mathbf{U} \in \mathbb{R}^{N \times n}$, $\Sigma \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{P \times n}$ are the truncated matrices of left singular vectors, singular values, and right singular vectors, respectively; \mathbf{E} , on the other hand, stands for the contribution of the discarded singular values (if any). We begin by partitioning \mathbf{X} into Q block matrices as follows

$$\mathbf{X} = [\mathbf{X}_1 \quad \mathbf{X}_2 \quad \cdots \quad \mathbf{X}_Q]. \quad (\text{B.2})$$

Next, we apply the SVD on each block matrix \mathbf{X}_i ($i = 1, 2, \dots, Q$), retaining only the first $r_i = \min(\text{rank}(\mathbf{X}_i), n)$ singular values:

$$\mathbf{X}_i = \mathbf{U}_i \Sigma_i \mathbf{V}_i^T + \bar{\mathbf{E}}_i \quad (i = 1, 2, \dots, Q) \quad (\text{B.3})$$

$\mathbf{U}_i \in \mathbb{R}^{N \times r_i}$, $\Sigma_i \in \mathbb{R}^{r_i \times r_i}$ and $\mathbf{V}_i \in \mathbb{R}^{P \times r_i}$ being the truncated matrices of left singular vectors, singular values, and right singular vectors, respectively (and $\bar{\mathbf{E}}_i$ the matrix corresponding to the discarded, if any, trailing singular values). Substitution of Eq. (B.3) into Eq. (B.2) yields

$$\mathbf{X} = [\mathbf{U}_1 \Sigma_1 \mathbf{V}_1^T \quad \mathbf{U}_2 \Sigma_2 \mathbf{V}_2^T \quad \cdots \quad \mathbf{U}_Q \Sigma_Q \mathbf{V}_Q^T] + \bar{\mathbf{E}}, \quad (\text{B.4})$$

where $\bar{\mathbf{E}} = [\bar{\mathbf{E}}_1 \quad \bar{\mathbf{E}}_2 \quad \cdots \quad \bar{\mathbf{E}}_Q]$. The above expression can be rearranged as follows

$$\begin{aligned} \mathbf{X} &= \underbrace{[\mathbf{U}_1 \Sigma_1 \quad \mathbf{U}_2 \Sigma_2 \quad \cdots \quad \mathbf{U}_Q \Sigma_Q]}_{\bar{\mathbf{X}}} \underbrace{\begin{bmatrix} \mathbf{V}_1^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{V}_Q^T \end{bmatrix}}_{\check{\mathbf{V}}^T} + \bar{\mathbf{E}} \\ &= \bar{\mathbf{X}} \check{\mathbf{V}}^T + \bar{\mathbf{E}}, \end{aligned} \quad (\text{B.5})$$

where $\bar{\mathbf{X}} \in \mathbb{R}^{N \times r}$ and $\check{\mathbf{V}} \in \mathbb{R}^{P \times r}$, with $r = \sum_{i=1}^Q r_i \leq nQ$. The final step consists in obtaining the truncated SVD of $\bar{\mathbf{X}}$ (of rank n)

$$\bar{\mathbf{X}} = \bar{\mathbf{U}} \bar{\mathbf{S}} \bar{\mathbf{V}}^T + \check{\mathbf{E}}, \quad (\text{B.6})$$

where $\bar{\mathbf{U}} \in \mathbb{R}^{N \times n}$, $\bar{\mathbf{S}} \in \mathbb{R}^{n \times n}$ and $\bar{\mathbf{V}} \in \mathbb{R}^{r \times n}$. Introducing the above factorization into Eq. (B.5), and moving the error terms to the left-hand side, we finally get

$$\mathbf{X} - (\bar{\mathbf{E}} + \check{\mathbf{E}}) = \bar{\mathbf{U}} \bar{\mathbf{S}} \mathbf{V}^{*T}, \quad (\text{B.7})$$

where $\mathbf{V}^{*T} := \bar{\mathbf{V}}^T \check{\mathbf{V}}^T$. Note that both $\check{\mathbf{V}}$ and $\bar{\mathbf{V}}$ are columnwise orthonormal matrices; it follows thus that \mathbf{V}^* exhibits also this property; indeed

$$\mathbf{V}^{*T} \mathbf{V}^* = \bar{\mathbf{V}}^T (\check{\mathbf{V}}^T \check{\mathbf{V}}) \bar{\mathbf{V}} = \bar{\mathbf{V}}^T (\mathbf{I}) \bar{\mathbf{V}} = \mathbf{I} \quad (\text{B.8})$$

($\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix). Since both $\bar{\mathbf{U}}$ and $\bar{\mathbf{S}}$ arise from an SVD, and therefore, are columnwise orthonormal and diagonal with positive entries, respectively, it follows from the uniqueness of such a decomposition that the factorization $\bar{\mathbf{U}} \bar{\mathbf{S}} \mathbf{V}^{*T}$ appearing in Eq. (B.7) is indeed the SVD of the matrix $\mathbf{X} - (\bar{\mathbf{E}} + \check{\mathbf{E}})$. Hence, $\bar{\mathbf{U}} \bar{\mathbf{S}} \mathbf{V}^{*T}$ can be regarded as an *approximation (of order $\mathcal{O}(\|\bar{\mathbf{E}} + \check{\mathbf{E}}\|)$) to the truncated SVD of rank n of matrix \mathbf{X}* . Notice that, if $n = \text{rank}(\mathbf{X})$, then $\bar{\mathbf{E}} = \check{\mathbf{E}} = \mathbf{0}$, and the described method would provide the exact factorization.

Concerning the memory saves provided by this approximated SVD, notice that it requires Q independent SVDs of matrices of size, on average, $N \times \frac{P}{Q}$, and one additional SVD over a matrix of size $N \times r$, with $r \leq nQ$. So, the larger matrix to be manipulated has only $r \leq nQ$ columns—recall that $n \ll P$ by hypothesis.

Appendix C. Elastic/inelastic dimensionality reduction

Let $\mathbf{X} \in \mathbb{R}^{N \times n}$ be a snapshot matrix, of either displacement, internal forces, or stresses, and let us decompose this matrix as $\mathbf{X} = [\mathbf{X}_e \quad \mathbf{X}_i]$, where \mathbf{X}_e and \mathbf{X}_i stand for the block matrices corresponding to solutions in the elastic and inelastic ranges, respectively. We seek an SVD-like factorization $\mathbf{X} = \bar{\mathbf{U}} \bar{\mathbf{S}} \bar{\mathbf{V}}^T$ (with $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$ orthonormal matrices, and $\bar{\mathbf{S}}$ diagonal with positive entries) such that the first n^{el} columns of $\bar{\mathbf{U}}$ form a basis matrix for the column space of the elastic matrix \mathbf{X}_e .

To this end, we first compute a generic orthogonal basis matrix $\mathbf{D} \in \mathbb{R}^{N \times n^{el}}$ for the column space of \mathbf{X}_e (using, for instance, the SVD itself), and then obtain the matrix arising from projecting \mathbf{X} onto the space spanned by \mathbf{D} :

$$\bar{\mathbf{X}}_e := \mathbf{D}(\mathbf{D}^T \mathbf{X}). \quad (\text{C.1})$$

In doing so, we can write

$$\mathbf{X} = \bar{\mathbf{X}}_e + \bar{\mathbf{X}}_i, \quad (\text{C.2})$$

where $\bar{\mathbf{X}}_i := \mathbf{X} - \bar{\mathbf{X}}_e$. Finally, introducing the SVD of both $\bar{\mathbf{X}}_e$ and $\bar{\mathbf{X}}_i$ in the preceding equation, we arrive at the desired factorization²²

$$\begin{aligned} \mathbf{X} &= \bar{\mathbf{U}}_e \bar{\mathbf{S}}_e \bar{\mathbf{V}}_e + \bar{\mathbf{U}}_i \bar{\mathbf{S}}_i \bar{\mathbf{V}}_i = \underbrace{\begin{bmatrix} \bar{\mathbf{U}}_e & \bar{\mathbf{U}}_i \end{bmatrix}}_{\bar{\mathbf{U}}} \underbrace{\begin{bmatrix} \bar{\mathbf{S}}_e & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{S}}_i \end{bmatrix}}_{\bar{\mathbf{S}}} \underbrace{\begin{bmatrix} \bar{\mathbf{V}}_e^T \\ \bar{\mathbf{V}}_i^T \end{bmatrix}}_{\bar{\mathbf{V}}^T} \\ &= \bar{\mathbf{U}} \bar{\mathbf{S}} \bar{\mathbf{V}}^T. \end{aligned} \quad (\text{C.3})$$

This strategy proves specially advantageous in those small strains, quasi-static problems in which the set of input parameters solely embodies variations of boundary conditions and external forces. Indeed, in such cases, the elastic response can be captured exactly by²³ $n^{el} = n_0 + n_f$ elastic modes and, therefore, by taking $n \geq n^{el}$, the reduced-order model is guaranteed to deliver elastic solutions with the same accuracy as the underlying (full-order) finite element model—in other words, only the solution in the inelastic range is subject to approximation.

²² Alternatively, when the sole variable of interest is the matrix of left singular vectors $\bar{\mathbf{U}}$, the inelastic modes may be obtained by simply calculating the SVD of $\mathbf{X}_i - \mathbf{D}(\mathbf{D}^T \mathbf{X}_i)$.

²³ Here, n_0 and n_f denote the number of parameters used to characterize the Dirichlet boundary conditions and external forces, respectively (see Section 3.2).

C.1. Modal analysis combined with SVD

In small strains, nonlinear vibration problems, a similar decomposition may offer some benefits when seeking dominant displacement modes; however, instead of employing as elastic modes the dominant left singular vectors arising from an SVD of the elastic snapshots, it proves more consistent to use the first n^{vib} natural vibrational modes (those with lowest frequencies), and determine the remaining modes using a “weighted SVD” (one that uses as minimization norm the mass matrix \mathbf{M}^h). The procedure can be sketched as follows: suppose that, after solving the corresponding eigenproblem, we have at our disposal the first n^{vib} natural vibration modes, denoted henceforth by Φ^{vib} (these modes are assumed to be \mathbf{M}^h -orthogonal). To determine the remaining $n^\perp = n - n^{vib}$ displacement modes, designated by Φ^\perp , we obtain first the Cholesky decomposition of the mass matrix \mathbf{M}^h , i.e., $\mathbf{M}^h = \bar{\mathbf{M}}^{hT} \bar{\mathbf{M}}^h$, and then compute the truncated SVD (of rank n^\perp) of the matrix defined as

$$\bar{\mathbf{X}} = \bar{\mathbf{M}}^h \left(\mathbf{X}_d - \Phi^{vib} (\Phi^{vibT} \mathbf{M}^h \mathbf{X}_d) \right). \quad (\text{C.4})$$

The desired basis matrix Φ^\perp finally emerges from making

$$\Phi^\perp = \bar{\mathbf{M}}^{h-1} \bar{\mathbf{U}}, \quad (\text{C.5})$$

$\bar{\mathbf{U}} \in \mathbb{R}^{N \times n^\perp}$ being the matrix of left singular vectors arising from the above mentioned SVD. It is easily seen that the resulting basis matrix for displacements

$$\Phi = [\Phi^{vib} \quad \Phi^\perp] \quad (\text{C.6})$$

is \mathbf{M}^h -orthonormal. Indeed, by definition, $\Phi^{vibT} \mathbf{M}^h \Phi^{vib} = \mathbf{I}$. Likewise, from Eq. (C.5), it follows that

$$\begin{aligned} \Phi^{\perp T} \mathbf{M}^h \Phi^\perp &= (\bar{\mathbf{M}}^{h-1} \bar{\mathbf{U}})^T (\bar{\mathbf{M}}^{hT} \bar{\mathbf{M}}^h) (\bar{\mathbf{M}}^{h-1} \bar{\mathbf{U}}) \\ &= \bar{\mathbf{U}}^T (\bar{\mathbf{M}}^{h-T} \bar{\mathbf{M}}^{hT}) (\bar{\mathbf{M}}^{h-1} \bar{\mathbf{M}}^h) \bar{\mathbf{U}} \\ &= \bar{\mathbf{U}}^T \bar{\mathbf{U}} = \mathbf{I}. \end{aligned} \quad (\text{C.7})$$

To complete the proof, it only remains to demonstrate that $\Phi^{\perp T} \mathbf{M}^h \Phi^{vib} = \mathbf{0}$. Since the column space of Φ^\perp is but a subspace of the column space of $\bar{\mathbf{M}}^{h-1} \bar{\mathbf{X}}$, this demonstration boils down to showing that $\Phi^{vibT} \mathbf{M}^h (\bar{\mathbf{M}}^{h-1} \bar{\mathbf{X}}) = \mathbf{0}$:

$$\begin{aligned} \Phi^{vibT} \mathbf{M}^h (\bar{\mathbf{M}}^{h-1} \bar{\mathbf{X}}) &= \Phi^{vibT} \mathbf{M}^h \left(\mathbf{X}_d - \Phi^{vib} (\Phi^{vibT} \mathbf{M}^h \mathbf{X}_d) \right) \\ &= \Phi^{vibT} \mathbf{M}^h \mathbf{X}_d - \overbrace{\Phi^{vibT} \mathbf{M}^h \Phi^{vib}}^{\mathbf{I}} (\Phi^{vibT} \mathbf{M}^h \mathbf{X}_d) \\ &= \mathbf{0}. \end{aligned} \quad (\text{C.8})$$

The advantage of this way of determining the displacement modes is that it ensures that the free, elastic vibration behavior corresponding to the first n^{vib} vibration modes is exactly captured—independently of the input parameters employed in the training process.

References

- [1] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, *Internat. J. Numer. Methods Engrg.* 86 (2) (2011) 155–181.
- [2] D. Ryckelynck, Hyper-reduction of mechanical models involving internal variables, *Internat. J. Numer. Methods Engrg.* 77 (1) (2009) 75–89.
- [3] M. Barrault, Y. Maday, N. Nguyen, A. Patera, An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations, *C. R. Math.* 339 (9) (2004) 667–672.
- [4] S. Chaturantabut, D. Sorensen, Discrete empirical interpolation for nonlinear model reduction, in: *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference, CDC/CCC 2009, Proceedings of the 48th IEEE Conference on, IEEE, 2010*, pp. 4316–4321.
- [5] N. Nguyen, A. Patera, J. Peraire, A best points interpolation method for efficient approximation of parametrized functions, *Internat. J. Numer. Methods Engrg.* 73 (2008) 521–543.

- [6] J. Baiges, R. Codina, S. Idelsohn, Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations, *Internat. J. Numer. Methods Fluids* 72 (12) (2013) 1219–1243.
- [7] P. Astrid, S. Weiland, K. Willcox, T. Backx, Missing point estimation in models described by proper orthogonal decomposition, *IEEE Trans. Automat. Control* 53 (10) (2008) 2237–2251.
- [8] R. Everson, L. Sirovich, Karhunen–Loeve procedure for gappy data, *J. Opt. Soc. Amer. A* 12 (8) (1995) 1657–1664.
- [9] K. Carlberg, J. Cortial, D. Amsallem, M. Zahr, C. Farhat, The GNAT nonlinear model reduction method and its application to fluid dynamics problems, in: 6th AIAA Theoretical Fluid Mechanics Conference, vol. 2730, Honolulu, Hawaii, June, 2011, pp. 2011–3112.
- [10] S. Chaturantabut, D.C. Sorensen, Application of POD and DEIM on dimension reduction of non-linear miscible viscous fingering in porous media, *Math. Comput. Model. Dyn. Syst.* 17 (4) (2011) 337–353.
- [11] P. Kerfriden, O. Gourg, T. Rabczuk, S.P.-A. Bordas, A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics, *Comput. Methods Appl. Mech. Engrg.* 256 (2013) 169–188.
- [12] M. Drophmann, B. Haasdonk, M. Ohlberger, Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation, *SIAM J. Sci. Comput.* 34 (2) (2012) A937–A969.
- [13] D. Galbally, K. Fidkowski, K. Willcox, O. Ghattas, Non-linear model reduction for uncertainty quantification in large-scale inverse problems, *Internat. J. Numer. Methods Engrg.* 81 (12) (2010) 1581–1608.
- [14] A. Radermacher, S. Reese, POD-based model reduction with empirical interpolation applied to nonlinear elasticity, *Internat. J. Numer. Methods Engrg.* (2015). URL <http://dx.doi.org/10.1002/nme.5177>.
- [15] M. Grepl, Y. Maday, N. Nguyen, A. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, *Math. Model. Numer. Anal.* 41 (3) (2007) 575–605.
- [16] H. Antil, S.E. Field, F. Herrmann, R.H. Nochetto, M. Tiglio, Two-step greedy algorithm for reduced order quadratures, *J. Sci. Comput.* 57 (3) (2013) 604–637.
- [17] J.A. Hernández, J. Oliver, A. Huespe, M. Caicedo, J. Cante, High-performance model reduction techniques in computational multiscale homogenization, *Comput. Methods Appl. Mech. Engrg.* 276 (2014) 149–189.
- [18] T. Aanonsen, Empirical Interpolation with Application to Reduced Basis Approximations, 2009.
- [19] S. An, T. Kim, D. James, Optimizing cubature for efficient integration of subspace deformations, *ACM Trans. Graph.* 27 (5) (2009) 165.
- [20] C. Farhat, P. Avery, T. Chapman, J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, *Internat. J. Numer. Methods Engrg.* 98 (9) (2014) 625–662.
- [21] C. Farhat, T. Chapman, P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, *Internat. J. Numer. Methods Engrg.* 102 (5) (2015) 1077–1110.
- [22] C. von Tycowicz, C. Schulz, H.-P. Seidel, K. Hildebrandt, An efficient construction of reduced deformable objects, *ACM Trans. Graph.* 32 (6) (2013) 213.
- [23] T. Kim, J. Delaney, Subspace fluid re-simulation, *ACM Trans. Graph.* 32 (4) (2013) 62.
- [24] Y. Teng, M.A. Otaduy, T. Kim, Simulating articulated subspace self-contact, *ACM Trans. Graph.* 33 (4) (2014).
- [25] C. Von-Tycowicz, C. Schulz, H.-P. Seidel, K. Hildebrandt, Real-time nonlinear shape interpolation, *ACM Trans. Graph.* 34 (3) (2015) 34.
- [26] J. Chadwick, S. An, D. James, Harmonic shells: a practical nonlinear sound model for near-rigid thin shells, *ACM Trans. Graph.* 28 (5) (2009).
- [27] S. Li, J. Huang, F. de Goes, X. Jin, H. Bao, M. Desbrun, Space-time editing of elastic motion through material optimization and reduction, *ACM Trans. Graph.* 33 (4) (2014) 108.
- [28] K. Carlberg, R. Tuminaro, P. Boggs, Preserving lagrangian structure in nonlinear model reduction with application to structural dynamics, *SIAM J. Sci. Comput.* 37 (2) (2015) B153–B184.
- [29] S. Boyaval, Reduced-basis approach for homogenization beyond the periodic setting, 2007. Arxiv preprint [math/0702674](https://arxiv.org/abs/math/0702674).
- [30] J. Yvonnet, Q. He, The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains, *J. Comput. Phys.* 223 (1) (2007) 341–368.
- [31] E. Monteiro, J. Yvonnet, Q. He, Computational homogenization for nonlinear conduction in heterogeneous materials using model reduction, *Comput. Mater. Sci.* 42 (4) (2008) 704–712.
- [32] N. Nguyen, A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales, *J. Comput. Phys.* 227 (23) (2008) 9807–9822.
- [33] Y. Efendiev, J. Galvis, E. Gildin, Local–global multiscale model reduction for flows in high-contrast heterogeneous media, *J. Comput. Phys.* 231 (24) (2012) 8100–8113.
- [34] Y. Efendiev, J. Galvis, F. Thomines, A systematic coarse-scale model reduction technique for parameter-dependent flows in highly heterogeneous media and its applications, *Multiscale Model. Simul.* 10 (4) (2012) 1317–1343.
- [35] A. Abdulle, Y. Bai, Reduced basis finite element heterogeneous multiscale method for high-order discretizations of elliptic homogenization problems, *J. Comput. Phys.* 231 (21) (2012) 7014–7036.
- [36] A. Abdulle, Y. Bai, Adaptive reduced basis finite element heterogeneous multiscale method, *Comput. Methods Appl. Mech. Engrg.* 257 (2013) 203–220.
- [37] L. Hogben, *Handbook of Linear Algebra*, Chapman & Hall/CRC, 2006.
- [38] W. Press, B. Flannery, S. Teukolsky, W. Vetterling, *Numerical Recipes in Fortran 77: The Art of Scientific Computing 1*, Cambridge University Press, Cambridge, 1993.
- [39] P. Krysl, S. Lall, J. Marsden, Dimensional model reduction in non-linear finite element dynamics of solids and structures, *Internat. J. Numer. Methods Engrg.* 51 (4) (2001) 479–504.
- [40] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge Univ Pr., 2004.
- [41] C.L. Lawson, R.J. Hanson, *Solving Least Squares Problems*, Vol. 161, SIAM, 1974.
- [42] J.C. Simo, T.J.R. Hughes, *Computational Inelasticity*, Springer, New York, 1998.
- [43] T. Belytschko, W.K. Liu, B. Moran, *Nonlinear Finite Elements for Continua and Structures*, John Wiley and Sons Ltd., New York, 2001.